НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:

Завідувач кафедри

Сергій СТІРЕНКО

«___»_____20__ р.

Дипломний проект

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Комп'ютерні системи та мережі»

спеціальності 123 «Комп'ютерна інженерія»

на тему: « Метод захищеної фільтрації зображень на віддалених комп'ютерних системах »

Виконав:
студент IV курсу, групи ІО-з61
Анар МАМЕДОВ _____

Керівник:
Доц. каф. ОТ, к. т. н.
Олександр МАРКОВСЬКИЙ _____

Консультант з нормоконтролю:
Професор, доктор технічних наук
Валерій СІМОНЕНКО _____

Рецензент:
Декан ФПМ, доктор технічних наук, професор
Іван ДИЧКА

_____

Засвідчую, що у цьому дипломному проекті немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2020 року

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» Факультет інформатики та обчислювальної техніки Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалаврський) Спеціальність – 123 «Комп'ютерна інженерія» Освітньо-професійна програма «Комп'ютерні системи та мережі»

ЗАТВЕРДЖУЮ
Завідувач кафедри
Сергій СТІРЕНКО
«___»_____2020 р.

ЗАВДАННЯ
на дипломний проект студенту
Анару МАМЕДОВУ

1. Тема проекту « Метод захищеної фільтрації зображень на віддалених комп'ютерних системах », керівник проекту Марковський Олександр Петрович, доцент, к. т. н., затверджені наказом по університету від «07» травня 2020 р. №1081-с

2. Термін подання студентом проекту 02.06.2020

3. Вихідні дані до проекту технічна документація

4. Зміст пояснювальної записки
Огляд методів захисту даних при їх віддаленій обробці.
Розробка методу захищеної середньоарифметичної фільтрації зображень в хмарах.
Розробка програм для моделювання віддаленої захищеної середньоарифметичної фільтрації зображень.

5. Перелік графічного матеріалу

Схема алгоритму, Принципова схема, Функціональна схема.

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Валерій СІМОНЕНКО		

7. Дата видачі завдання

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	*Затвердження теми роботи*		
2.	*Вивчення літератури*	20.12.2019-14.01.2020	
3.	*Створення та узгодження технічного завдання*	15.01.2020-26.01.2020	
4.	*Розробка методу захищеної середньоарифметичної фільтрації зображень в хмарах*	27.01.2020-14.02.2020	
5.	*Розробка програмної моделі*	15.02.2020-30.04.2020	
6.	*Відлагодження програми та виправлення помилок*	01.05.2020-15.05.2020	
7.	*Оформлення документації дипломного проєкту*	16.05.2020-06.06.2020	
8.	*Захист програмного продукту*	18.05.2020	
9.	*Передзахист*	24.05.2020	
10.	*Захист*		

Студент Анар МАМЕДОВ

Керівник Олександр МАРКОВСЬКИЙ

Анотація

Ціллю представлених в дипломному проекті досліджень є підвищення ефективності захищеної фільтрації потоків зображень на віддалених комп'ютерних системах з використанням хмарних технологій за рахунок збільшення рівня їх захищеності в процесі віддаленої обробки при близьких до теоретичного мінімуму витратах ресурсів користувача на реалізацію функцій захисту.

Розроблено та досліджено новий метод захищеної середньоарифметичної фільтрації зображень на віддалених комп'ютерних системах. Запропонований метод дозволяє надійно захистити зображення від їх реконструкції методами статистичного аналізу, забезпечуючи при цьому прискорення фільтрації на 1-2 порядки.

Ключові слова: середньоарифметична фільтрація, захищені обчислення, захищена обробка зображень, хмарні обчислення, хмарні технології.

Abstract

The goal of presented by diploma project research is to improve the efficiency of image stream filtration using remote computer systems under cloud technologies. Researches stipulate that the aim can be achieved by enhancing image security during remote processing with the minimum cost of a user`s resources to implement security features.

The new method of arithmetical mean image filtration with the usage of remote computer systems has developed. The proposed method ensures the acceleration filtering by 1-2 orders of magnitude

Keywords: arithmetic mean filtration, secure computing, secure image processing, cloud computing, cloud technologies.

Аннотация

Целью представленных в дипломном проекте исследований является повышение эффективности защищенной фильтрации потоков изображений на удаленных компьютерных системах с использованием облачных технологий за счет повышения уровня их защищенности при близких к теоретическому минимуму затратах ресурсов пользователя на реализацию функций защиты. Разработан и исследован новый метод защищенной среднеарифметической фильтрации на удаленных компьютерных системах. Предложенный метод позволяет надежно защитить изображения от их восстановления методами статистического анализа, обеспечивая при этом ускорение фильтрации на 1-2 порядка.

Ключевые слова: среднеарифметическая фильтрация, защищенные вычисления, защищенная обработка изображений, облачные вычисления, облачные технологии.

Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість сторінок	№ екземпляр	Примітка
		Документація загальна			
		Розроблена заново			
А4	ІАЛЦ.468243.002 ТЗ		5		
		Метод захищеної фільтрації зображень на віддалених комп’ютерних системах			
		Технічне завдання			
А4	ІАЛЦ.468243.003 ПЗ				
		Метод захищеної фільтрації зображень на віддалених комп’ютерних системах			
А3	ІАЛЦ.468243.004 Э1	Пояснювальна записка			
			1		
		Схема функціональна			
А3	ІАЛЦ.468243.005 Э3		1		
		Схема принципова			
А4	ІАЛЦ.468243.006 Д1		1		
		Схема алгоритму			

Зм.	Арк.	№ докум.	Підпис	Дата	ІАЛЦ.468243.001 ОА		
Розроб.		Мамедов А.Р.			Метод захищеної фільтрації зображень на віддалених комп’ютерних системах Опис альбому	Літ.	Аркуш 1 / Аркушів 1
Перевір.		Марковський О.П.					
Н. контр.		Сімоненко В.П.				НТУУ «КПІ» ФІОТ Група ІО-з61	
Затв.							

Технічне завдання

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Технічне завдання поширюється на розробку нового методу захищеної фільтрації зображень на віддалених комп’ютерних системах.

Область застосування – системи обробки аерокосмічних зображень, зображень з моніторів відео спостереження, зображень в засобах технічного зору робото технічних комплексів.

2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання роботи кваліфікаційно-освітнього рівня “бакалавр комп’ютерної інженерії”, затверджене кафедрою обчислювальної техніки Національного технічного університету України “Київський політехнічний інститут імені Ігоря Сікорського”.

3. МЕТА ТА ПРИЗНАЧЕННЯ РОЗРОБКИ

Мета проекту полягає в підвищенні ефективності захищеної обробки зображень, зокрема їх середньоарифметичної фільтрації на віддалених обчислювальних потужностях в рамках хмарних технологій за рахунок збільшення питомої ваги об’єму обчислень, що виконуються на них і, відповідно, зменшення долі обчислень, що реалізуються користувачем.

Розробка призначена для прискорення обробки зображень за рахунок безпечного в інформаційному плані залучення віддалених багато-процесорних комп’ютерних систем з використанням сучасних хмарних технологій.

4. ДЖЕРЕЛА РОЗРОБКИ

4.1. Марковський О.П. Метод прискореної захищеної фільтрації зображень на віддалених комп'ютерних системах / О.П. Марковський, І.О. Гуменюк., Міратаї Аліреза, Я.І. Торошанко, М.О.Волощук // Телекомунікаційні та інформаційні технології**.** - № 4 (65).- 2019.- С.99-110.

4.2. Гуменюк І.О. Метод віддаленої середньоарифметичної фільтрації зображень / І.О. Гуменюк, О.Є.Слюсаренко // Альманах науки. - 2019.- № 11 (32).- С.40-43.

4.3. Марковський О.П. Захищена реалізація фільтрації зображень в GRID-системах / О.П. Марковський, М.В. Невдащенко, А.М. Білашевська // Вісник Національного технічного університету України "КПІ" Інформатика, управління та обчислювальна техніка. – Київ: ВЕК+. – 2014. – № 61. – С.105-109.

4.4 Monjur Ahmed. Cloud Computing and Security Issues in the Cloud / Monjur Ahmed, Mohammad Ashraf Hossain // International Journal of Network Security PengyaoWang. Rapid processing of remote sensing images based on cloud computing /PengyaoWang,JianqinWang, YingChen, Guangyuan Ni // Future Generation Computer Systems. – Vol.29.- № 8.-2013.- pp.1963-1968.

4.5 Sutton M. A. Image Correlation for Shape, Motion and Deformation Measurements (Basic Concepts, Theory and Applications)./ Sutton M. A., Orteu J. J., Schreier H. // – New York: Springer, 2009. – 364 p.

5. ТЕХНІЧНІ ВИМОГИ

5.1. Вимоги до продукту, що розробляється

5.1.1 Метод фільтрації зображень – середньоарифметична фільтрація з використанням квадратної апертури : середній елемент апертури при фільтрації замінюється на середнє арифметичне всіх точок апертури.

5.1.2 Метод має забезпечувати прискорення фільтрації зображення за рахунок залучення віддалених обчислювальних потужностей в рамках хмарних не менше ніж в десять раз в порівнянні з реалізацією фільтрації на обчислювальній платформі користувача.

5.1.3 Розмір апертури: 11 x 11, 13 x 13, 17 x 17.

5.1.4 Метод має забезпечувати надійний захист зображення від незаконного доступу до нього в процесі його передачі на віддаленні комп'ютерні системи а також в процесі безпосередньої обробки.

5.1.5 Об'єм перебору ключів для отримання незаконного доступу до зображення в процесі його обробки має виходити за рамки технічних можливостей сучасних комп'ютерних систем.

5.2. Вимоги до програмного забезпечення

- Операційна система MS Windows 10
- Visual Studio 2017
- C++11

5.3. Вимоги до апаратної частини

- Процесор рівня Intel i5 і вище.
- Оперативна пам'ять не менше 500 МБ.
- Вільне місце на жорсткому диску не менше 100 МБ.

6. ЕТАПИ РОЗРОБКИ

	Дата
Вивчення літератури	20.12.2019
Створення та узгодження технічного завдання	15.01.2020

Вивчення літературних джерел	27.01.2020
Розробка методу ффльтрації	14.02.2020
Розробка програмної моделі	01.05.2020
Відлагодження програми та виправлення помилок	15.05.2020
Оформлення документації дипломного проекту	06.06.2020

Пояснювальна записка

ЗМІСТ

ВСТУП

Ще в 70-х роках минулого століття була описана ідея створення великих центрів обробки даних з можливістю віддаленого доступу до них [1]. На початку нового тисячоліття, завдяки стрімкому розвитку комп'ютерних мереж та багатопроцесорних систем, концепція хмарних технологій почала втілюватись в життя.

Завдячуючи цій прогресивній концепції, широкі кола користувачів отримують доступ до практично необмежених комп'ютерних ресурсів. Це дозволяє вирішувати перебором ті задачі, які теоретики вважали нерозв'язними. Також, хмарні технології дозволяють постійно та рівномірно навантажувати багатопроцесорні системи, що робить їхнє створення істотно рентабельнішим та оптимізує використання обчислювальних потужностей в цілому. Тобто, хмарні технології відкривають принципово нові можливості в вирішення як практичних, так наукових задач.

Попри наведенні можливості хмарних технологій, до теперішнього часу розповсюдження набули лише хмарні сховища даних. Основною перешкодою широкого використання хмарних обчислень є незахищеність даних в процесі їх обробки на непідконтрольних користувачам віддалених обчислювальних потужностях. Існує великий ризик викрадення чи спотворення даних під час їхньої обробки на віддалених непідконтрольних користувачу обчислювальних потужностях. Разом з тим, існує такий клас задач, в якому конфіденційність даних є дуже важливою.

До класу таких задач повною мірою можна віднести задачі обробки зображень. Характерною ознакою розвитку інформаційних технологій на межі тисячоліть є якісне вдосконалення інтерфкйсу між компютерними системаи

Зокрема, однією з них є обробка аерокосмічних знімків, обробка результатів відео спостережень, обробка зображень персон перед їх

розпізнаванням в системах відео контролю. Для всіх цих практично важливих застосувань доступ до зображень під час їх обробки на віддалених комп'ютерних системах з боку сторонніх осіб є недопустимим [2].

Для захисту інформації в процесі її обробки, пов'язаної з її зміною та перетвореннями, потрібно застосовувати криптографічне шифрування, яке залежить від виду перетворень. Відповідно, для захисту зображень в процесі їх середньоарифметичної фільтрації потрібно розробити спеціальні, гомоморфні дл процедури фільтрації методи шифрування зображень та дешифрування відфільтрованого зображення.

Таким чином, створення криптографічних механізмів захисту зображень в процесі їхньої обробки – середньоарифметичної фільтрації на віддалених обчислювальних потужностях в рамках хмарних технологій є актуальним на сучасному етапі розвитку інформаційних технологій.

РОЗДІЛ 1

ОГЛЯД МЕТОДІВ ЗАХИСТУ ДАНИХ ПРИ ЇХ ВІДДАЛЕНІЙ ОБРОБЦІ.

Поява та швидке поширення хмарних технологій надає змогу широкому колу користувачів залучати значні обчислювальні потужності для вирішення своїх прикладних задач. Це дозволяє перейти на якісно новий рівень вирішення наукових, проектних та економічних задач. Оптимізація значної частини моделей може бути виконана методами перебору чи направленого перебору [1]. Таким чином, хмарні технології виступають могутнім чинником прискорення технологічного прогресу практично в усіх сферах людської діяльності.

Широкому впровадженню використанню віддалених обчислювальних потужностей заважає наявність реальної загрози доступу до даних користувачів безпосередньо в процесі їх обробки на віддалених і непідконт-рольних обчислювальних системах. Тому постає задача створення методі і засобів захисту інформації користувачів в процесі їх обробки. Мова йде про створення систем шифрування вихідних даних, результати яких можуть безпечно передаватися для віддаленої обробки і дешифруватися користувачем. при цьому існує суттєве обмеження на об’єм ресурсів які можуть бути використані для реалізацію функцій захисту інформації: вони мають бути на порядки меншими ніж об’єм обчислювальних ресурсів, що витрачається на обробку цієї інформації.

1.1 Аналіз вимог до засобів захисту даних при їх віддаленій обробці

При визначенні критеріїв ефективності засобів захисту інформації користувачів при їх віддаленій обробці доцільно на системному рівні виходити із основних положень ефективності комерційної криптографії:

- рівень захищеності даних визначається об’ємом ресурсів, які має витрати сторона, яка здійснює порушення захисту для того, щоб отримати незаконний доступ до даних. Ці ресурси можуть включати власне обчислювальні ресурси, ресурси пам'яті, часові ресурси, інформаційні ресурси;

- реалізація засобів захисту також потребує певних ресурсів, які має витратити користувач, щоб зашифрувати дані та дешифрувати отримані від віддаленої системи результати обробки цих даних;

- рівень захищеності даних з використанням певних засобів має бути таким, щоб порушення захисту, тобто отримання незаконного доступ до даних було економічно недоцільним. Це означає, що вигода від незаконного отримання доступу до інформації має бути меншою вартості ресурсів, що які має витрати сторона, яка здійснює порушення захисту для доступу до закритих даних користувача;

- затрати користувача на реалізація засобів захисту також потребує певних ресурсів, які має витратити користувач, щоб зашифрувати дані та дешифрувати отримані від віддаленої системи результати обробки цих даних мають бути меншими ніж втрати від незаконного доступу до інформації з боку сторонніх осіб.

Наведені фундаментальних принципи визначають наступні загальні критерії ефективності засобів захисту інформації користувачів:

- рівень захищеності даних, який забезпечується первними криптографічними перетвореннями над даними. Оцінкою значення цього критерію є витрати обчислювальних ресурсів зловмисником для отримання незаконного доступу до даних, тобто для порушення створеного засобами захисту;

- співвідношення числа операцій, що виконуються на обчислювальній платформі користувача при використанні віддалених обчислень і числа операцій за умови, що весь об'єм операцій по обробці даних здійснюється користувачем на власній обчислювальній платформі;

- можливості організації ефективного обчислювального процесу для реалізації функцій захисту на спеціалізованих і проблемно-орієнтованих апаратних засобах;

- стійкість зашифрованих даних до негативного впливу навмисних і ненавмисних перешкод у лініях передачі даних.

Наведені критерії є суперечливими за своїм змістом: покращення одного з них має наслідком погіршення інших, так, що інтегральним показником якості засобів захисту інформації можна вважати ступінь компромісу між цими критеріями відповідно конкретних умов застосування, які визначають значимість вказаних критеріїв.

Виходячи з цих фундаментальних принципів організації захисту даних користувачів можна конкретизувати критерії ефективності для систем гомомогенного шифрування даних при використанні віддалених незахищених комп'ютерних систем в рамках хмарних технологій.

Сутність гомомогенного шифрування даних *D* схематично полягає в наступному:

Вважається, що користувачеві потрібно виконати деяку обробку даних D, яка може бути позначена через *f*(*D*) з отриманням результату R:

$$R = f(D). \qquad (1.1)$$

Для прискорення обробки даних, користувач використовує віддалені обчислювальні потужності, які мають в своєму складі велику кількість процесорів і здані значно прискорити реалізацію перетворення (1.1) головним чином, за рахунок розпаралелювання відповідного обчислювального процесу.

Для того, об захистити дані D користувач шифрує ці дані з використанням певного криптографічного шифру, який здійснює над даними перетворення C, в результаті якого отримує зашифровану дані, які можуть бути позначені як $C(D)$. Таким чином, користувач надсилає у хмару зашифровані дані $C(D)$. Хмара виконує пошук вільних обчислювальних потужностей у відповідності з затребуваними ресурсами для виконання перетворень f і пересилає зашифровані дані $C(D)$ на вибрану віддалену комп'ютерну систему. Система здійснює обробку інформації згідно визначеної процедури з отриманням результату $Q = f(C(D))$. Результат обчислень Q пересилається через хмару користувачеві. Останній здійснює дешифрування отриманих результатів з використанням певних криптографічних перетворень G і отримує результат $G(Q)$. Криптографічні перетворення, які виконує користувач називаються гомоморфними, якщо виконується наступна умова:

$$G(f(C(D))) = R = f(D). \qquad (1.2)$$

Іншими словами, в результаті застосування гомоморфного шифрування користувач має отримати користувач має отримати коректний результат обробки інформації.

Таким чином, для ефективного шифрування даних користувачів при їх віддаленій обробці в рамках хмарних технологій мають використовуватися гомоморфні методи: тобто процедура шифрування визначається морфологією тих перетворень над інформацією користувача, які виконуються в процесі її віддаленої обробки і вона має забезпечити можливість адекватного дешифрування результатів обробки.

Конкретизуючи наведені вище загальні принципи організації захисту інформації можна дійти до висновку, що гомоморфне шифрування має забезпечувати такий рівень захисту даних, щоб ресурси для подолання захисту були більшими за потенціальні вигоди від незаконного доступу до інформації користувачів. При цьому існує суттєве обмеження на об’єм

ресурсів які можуть бути використані для реалізацію функцій захисту інформації: вони мають бути на порядки меншими ніж об'єм обчислювальних ресурсів, що витрачається на обробку цієї інформації.

Теоретично [4], в основі будь-якого механізму криптографічного захисту даних покладене незворотне математичне перетворення, тобто перетворення для якого визначено процедуру прямого перетворення, з якого не можна отримати процедуру зворотного перетворення [5]. В арсеналі сучасній засобів криптографічного захисту інформації користувачі застосовується декілька типів таких незворотних математичних перетворень. Зокрема, в хеш-алгоритмах, потокових алгоритмах шифрування, алгоритмах симетричного шифрування використовуються незворотні перетворення на основі нелінійних булевих перетворень. В булевій алгебрі мжна визначити систему ортогональних булевих функцій, що трансформують вхідний вектор у вихідний. Якщо булеві функції цього перетворення нелінійні, теоретично доведено [6], що не існує методу аналітичного отримання зворотного заданому перетворення, тобто не існує методів аналітичного розв'язання систем нелінійних булевих рівнянь. Основна перевага використання булевих незворотних перетворень полягає в тому, що вони забезпечують можливості для швидкої програмної та апаратної реалізації [7]. Основний недолік побудови засобів захисту даних з використанням незворотних булевих перетворень полягає в вузьких функціональних здатностях, що не дозволяє будувати на їх основі складні конструкції криптографічного захисту .

1.2 Огляд технологій захисту даних при їх віддаленій обробці

Суттєвий недолік хмарних технологій, який істотно обмежує їх практичне застосування в плані доступу користувачів саме до

обчислювальних ресірсів, полягає в потенціальній загрозі незаконного доступу сторонніх особ до даних користувача при їх передачі та обробці на віддалених обчислювальних потужностях. Положення з захистом інформації при віддалені обробці даних суттєвим чином різниться у порівнянні з віддаленим зберіганням інформації, яке також забезпечуються в рамках хмарних технологій. В останньому випадку проблема захисту інформації користувачів при її зберіганні на непідконтрольних ним віддалених носіях вирішується значно простіше – використанням стандартизованих надійних симетричних шифрів типу AES. Це означає, що дані користувачів, залежно від вимог до рівня їх захищеності, шифруються з використанням симетричних шифрів типу AES, DES або потокового шифрування [13]. Шифрування та дешифрування здійснюється на процесорних засобах користувача. Це забезпечує виконання принципу єдиного володаря секретного ключа і виключає незаконний доступ до інформації користувачів під в процесі зберігання їх даних на віддалених носіях, а також непомітне для користувачів внесення змін до даних.

Зовсім інша картина має місце при захисті даних користувачів в процесі їх безпосередньої обробки на віддалених і, відповідно, неконтрольованих, обчислювальних системах. До теперішнього часу проблема збереження в секреті даних, що оброблюються на віддалених системах, не вирішена в силу великої її складності [14].

Проблемі захисту даних користувача в системах віддаленої обробки присвячено в останні роки значне число робіт [15,16]. Основна складність полягає в тому, що не існує і не може існувати єдиного підходу до захисту даних у процесі їх обробки. Реально, переважна частина виконаних досліджень вирішує проблему захисту даних тільки для окремих класів обчислювальних задач, наприклад, для лінійної алгебри, обробки зображень, матричних перетворень, модулярного експоненціювання, перетворень на кінцевих полях Галуа [15].

Основною проблемою, на вирішення якої направлені виконані до теперішнього часу дослідження присвячені захисту даних безпосередньо під час їх обробки, полягає в використанні спеціальних методів шифрування, гомоморфних перетворенням, які виконуються над данимми в процесі їх віддаленої обробки. Ці методи шифрування-дешифрування дозволяють отримувати коректний результат шляхом дешифрування результатів віддаленої обробки зашифрованих даних. З наведенного випливає, що не існує універсальних методів шифрування даних перед їх віддаленою обробкою, які не залежать від операцій обробки даних користувачів. Це практично означає, що для кожного виду віддаленої обробки даних користувачів потрібно окремо розробляти гомоморфний метод шифрування та дешифрування даних.

Зокрема, в роботі [15] запропоновано методи захищеної реалізації масових операцій обробки зображень - медіанної та середньоарифметичної фільтрації, які використовуються для поліпшення якості зображень. В основі цих методів покладено інтервальне шифрування точок зображення перед передачею його для обробки в хмарних системах. Метод забезпечує виконання медіанної фільтрації на віддалених відкритих комп'ютерних системах, закриваючи при цьому доступ до справжнього зображення.

В роботі [16] запропоновано методи захищеної реалізації масових операцій обробки сигналів – прямого та зворотного перетворення Фур'є при їх реалізації на віддалених обчислювальних потужностях в рамках хмарних технологій. В цих методах використовується адитивне потокове шифрування. Це абезпечує високу швидкодію і орієнтовано на потокових характер обробки сигналів користувача.

Реально опубліковані до теперішнього часу дослідження, які стосуються захисту даних при їх віддаленій обробці, можна підрозділити на два класи. До першого класу відносяться роботи, котрі для окремих класів видів віддаленої обробки вирішуються задачу унеможливленя незаконного

доступу до даних, які обробляються віддалено на непідконтрольних обчислювальних потужностях. Тобто тут мова йде саме про гомоморфне шифрування та дешифрування даних в їх класичному розумінні.

Другий класс задач захисту даних при їх віддаленій обробці складають дослідження направлені на створення механізмів контролю правильності обчислень, що виконуються на віддалених комп'ютерних системах [16]. Така задача постає для широкого кола практичних застосувань, оскільки в процесі віддаленої обробки може бути отримано неправильний результат під дією спеціального або ненавмисного впливу на процес обробки даних користувачів.

Інтенсивно проводяться дослідження, спрямовані на створення ефективної захищеного виконання на потенційно відкритих віддалених комп'ютерних системах операції модулярного експоненціювання - базової процедури широкого класу протоколів захисту інформації [17]. Стимулом до віддаленої реалізації цієї базової і ресурсоємкої операції захисту інформації стало розширене використання компактних малопотужних термінальних пристроїв комп'ютерних систем управління та моніторингу об'єктів реального світу. Такі термінальні процесори мають вбудований радіомодем і орієнтовані на передачі даних в мережі Інтернет [18]. Відповідно, вони підтримують існуючі протоколи мережевої безпеки, які основані на використанні операції модулярного експоненціювання. Разом, з тим недостатня обчислювальна потужність таких портативних термінальних пристроїв не дозволяє виконувати відповідні обчислення достатньо швидко.

Аналіз існуючих методів захищеного модулярного експоненціювання показав, що рішення задачі гомоморфного шифрування полягає в розділенні обчислень на дві частини - одна, велика частина, виконується на віддалених комп'ютерних системах, інша, менша за обсягом – на обчислювальній платформі користувача.

Цей критерій дозволяє проводити оцінку можливість прискорення реалізації мережевих протоколів захисту інформації, в основі яких лежать операції модулярного експоненціювання, за рахунок залучення можливостей сучасних хмарних технологій без шкоди для інформаційної безпеки.

У роботі [18] запропоновано метод віддаленого обчислення модулярної експоненти на основі випадкового поділу коду експоненти E на групи розрядів. Це дозволяє організувати обчислення модулярної експоненти $A^E \mathrm{mod} M$ у вигляді добутків окремих можулярних експонент. Ці часткові модулярні експоненти можуть обчислюватися незалежно на паралельних процесорах віддалених комп'ютерних систем. Формування добутку, тобто формування результату $A^E \mod M$ здійснюється безпосередньо на обчислювальній платформі користувача. Для захисту компоненти A – числа, над яким здійснюється операція модулярнеекспонеціювання, застосовується виконання декількох перших кроків експоненціювання також на обчислювальній платформі користувача.

Зрима перевага розглянутого методу полягає в тому, що у повній мірі можуть бути використані можливості багатопроцесорних віддалених систем із паралельного обчислювання окремих експонент. У роботі [18] на основі теоретичних і експериментальних даних встановлено, що описаний метод дозволяє приблизно у три рази прискорити реалізацію операції модулярного експоненціювання.

Показано, що невирішеність проблеми забезпечення інформаційної безпеки безпосередньо в процесі її обробки на віддалених і не контрольованих потужних обчислювальних систем з використанням хмарних технологій суттєвим чином стримує широке використання переваг, які надають ці технології і це педалює інтенсивні дослідження,

направлені на створення ефективних засобі захисту даних користувачів при їх віддаленій обробці.

До теперішнього часу створено і успішно використовується на практиці низка методів гомоморфного шифрування для задач перетворень матриць, лінійної алгебри, розпізнавання образів, модулярного експоненціювання.

1.3 Аналіз існуючих технологій захисту зображень при їх віддаленій обробці

Обробка зображень та сигналів відноситься до класу найбільш масових операцій. Розвиток інформаційних технологій характеризується динамічним вдосконаленням інтерфейсом між об’єктами реального світу та комп’ютерами. Дуже значну роль в цьому процесі відіграють системи обробки зображень на сигналів.

Загальною рисою задач обробки зображень є їх висока ресурсоємкість: сучасні зображення містять мільйони точок. практика сучасного використання комп’ютерного аналізу зображень потребує значного збільшення об’єму обчислювальних ресурсів, темпи росту яких помітно випереджають прогрес потужності процесорів. Найбільш перспективним з наразі існуючих шляхів прискорення обробки зображень є залучення віддалених обчислювальних ресурсів в рамках хмарних технологій.

На сьогодні найбільшої перешкодою на шляху широкого використання переваг хмарних технологій для залучення практично необмежених обчислювальних потужностей для обробки зображень виступає їх незахищеність від стороннього доступу.

Тобто, існує необхідність створення механізмів захисту зображень від несанкціонованого доступу в процесі їх віддаленої обробки.

На сьогоднішній день запропоновано ряд підходів для вирішення цієї важливої для практики задачі. Задачі обробки зображень являють собою одну із найбільш масових процедур сучасних інформаційних технологіях,

які добре розпаралелюються і тому можуть ефективно вирішуватися на віддалених багатоядерних обчислювальних системах з залученням хмарних технологій. Більша частина зображень, які потенційно можуть оброблюватися на віддалених потужних комп'ютерних системах мають конфіденційних характер.

Одним з найбільш розповсюджених процедур попередньої обробки зображень є поліпшення їх якості за допомогою фільтрації. В процесі формування зображення на відеосистемах та при передачі їх по ефірним каналм виникають імпульсні завади, які негативним чином впливають на якість зображення і стають завадою для подальшого опрацювання зображення: актуалізації їх елементів та розпізнавання. Для поліпшення якості зображень шляхом усунення імпульсних завад найчастіше використовують медіанну та середньоарифметичну фільтрацію.

Сутність медіанної фільтрації полягає в тому що, середній елемент поточної апертури замінюється середній елементом відсортованої послідовності елементів поточної апертури. Зміст середньоарифметичної фільтрації полягає в тому що, середній елемент поточної апертури замінюється середній елементом відсортованої послідовності елементів поточної апертури.

До сьогоднішнього часу розроблено ряд рішень задачі захисту зображень в процесі їхньої віддаленої середньоарифметичної фільтрації.

В роботі [15] запропоновано методи захищеної реалізації масових операцій обробки зображень - медіанної та середньоарифметичної фільтрації, які використовуються для поліпшення якості зображень. В основі цих методів покладено інтервальне шифрування точок зображення перед передачею його для обробки в віддалених системах. Розглянутий метод забезпечує виконання медіанної фільтрації на віддалених відкритих комп'ютерних системах, закриваючи при цьому доступ до справжнього зображення.

Суть іншого із відомих методів [20] захищеної реалізації середньоарифметичної фільтрації полягає у додаванні оригінального зображення P матриці F випадкових цілих чисел, кожен елемент f_{ij} якої є кратним m: $f_{ij} \bmod m = 0$, де m – просте натуральне число, а $\forall i \in \{1, 2, ..., k\},\ j \in \{1, 2, ..., h\} : p_{ij} < m$. Захищене таким чином зображення B надсилається на віддалену систему для виконання середньоарифметичної фільтрації. Отримане відфільтроване в віддлений комп'ютерній системі зображення C дешифрується шляхом ділення кожного його елементу c_{ij} за модулем m, з отриманням відфільтрованого оригінального зображення Q^1 у наступному вигляді: $q_{ij}^1 = c_{ij} \bmod m$ [20].

Основний недолік цього відомого методу [20] полягає є високій обчислювальній складності операції дешифрування отриманого від віддаленої системи зображення , а саме використання повільної операції ділення для знаходження остачі від ділення за модулем m для кожної точки зображення.

Іншим відомим рішенням [21] задачі захисту зображень в процесі їх передачі та обробки на віддалених комп'ютерних системах полягає в застосуванні зображення шляхом додавання до матриці пікселів зображення матриці L випадкових чисел до матриці P оригінального зображення, з отриманням захищеного зображення D. Це захищене зображення D надсилається для фільтрації на віддалену обчислювальну систему. Дешифрування здійснюється по такій схемі: від повернутою системою матриці віднімається матриця відфільтрованого зображення масок, які використовувалися для шифрування. Метод передбачає, що матриці масок заздалегідь формуються користувачем з залученням власних обчислювальних ресурсів та використовуються повторно.

Основним недоліком цього методу є низький рівень захищеності від спроб отримання незаконного доступу до зображення застосуванням зловмисником статистичного аналізу.

В роботі [16] запропоновано методи захищеної реалізації масових операцій обробки сигналів – прямого та зворотного перетворення Фур'є при їх реалізації на віддалених обчислювальних потужностях в рамках хмарних технологій. В цих методах використовується адитивне потокове шифрування. Це забезпечує високу швидкодію і орієнтовано на потокових характер обробки сигналів користувача.

Таким чином, показано, що існуючі методи захисту зображень при їх віддаленій фільтрації базуються на адитивному маскуванні і не забезпечують прийнятної для потреб сьогоднішнього дня рівня ефективності. В якості найбільш перспективних шляхів вдосконалення методів захищеної віддаленої фільтрації зображень доцільно розглядати урахування того, що на практиці частіше за все оброблюються потік зображень, які можуть перемішувати і таким чином утруднювати реконструкцію кожного конкретного зображення при їх віддаленій обробці.

ВИСНОВКИ ДО РОЗДІЛУ 1

В результаті проведених студій та досліджень, які складають оглядовий розділ дипломного проекту і мали на меті аналіз існуючих технологій криптографічного захисту даних при їх віддаленій обробці, виявлення можливостей їх застосування для захисту зображень при їх віддаленій обробці і, зокрема, середньоарифметичній фільтрації, можна зробити такі висновки:

1. Показано, що невирішеність проблеми забезпечення інформаційної безпеки безпосередньо в процесі її обробки на віддалених і не контрольованих потужних обчислювальних систем з використанням хмарних технологій суттєвим чином стримує широке використання переваг, які надають ці технології і це педалює інтенсивні дослідження, направлені на створення ефективних засобі захисту даних користувачів при їх віддаленій обробці.

2. Для ефективного шифрування даних користувачів при їх віддаленій обробці в рамках хмарних технологій мають використовуватися гомоморфні методи: тобто процедура шифрування визначається морфологією тих перетворень над інформацією користувача, які виконуються в процесі її віддаленої обробки і вона має забезпечити можливість адекватного дешифрування результатів обробки. Гомоморфне шифрування має забезпечувати такий рівень захисту даних, щоб ресурси для подолання захисту були більшими за потенціальні вигоди від незаконного доступу до інформації користувачів. При цьому існує суттєве обмеження на об’єм ресурсів які можуть бути використані для реалізацію функцій захисту інформації: вони мають бути на порядки меншими ніж об’єм обчислювальних ресурсів, що витрачається на обробку цієї інформації.

3. До теперішнього часу створено і успішно використовується на практиці низка методів гомоморфного шифрування для задач перетворень матриць, лінійної алгебри, розпізнавання образів, модулярного експоненціювання.

4. Задачі обробки зображень являють собою одну із найбільш масових процедур сучасних інформаційних технологіях, які добре розпаралелюються і тому можуть ефективно вирішуватися на віддалених багатоядерних обчислювальних системах з залученням хмарних технологій. Більша частина зображень, які потенційно можуть оброблюватися на віддалених потужних комп'ютерних системах мають конфіденційних характер.

5. Існуючі методи захисту зображень при їх віддаленій фільтрації базуються на адитивному маскуванні і не забезпечують прийнятної для потреб сьогоднішнього дня рівня ефективності. В якості найбільш перспективних шляхів вдосконалення методів захищеної віддаленої фільтрації зображень доцільно розглядати урахування того, що на практиці частіше за все оброблюються потік зображень, які можуть перемішувати і таким чином утруднювати реконструкцію кожного конкретного зображення при їх віддаленій обробці.

РОЗДІЛ 2

РОЗРОБКА МЕТОДУ ЗАХИЩЕНОЇ СЕРЕДНЬОАРИФМЕТИЧНОЇ ФІЛЬТРАЦІЇ ЗОБРАЖЕНЬ В ХМАРАХ

Середньоарифметична фільтрація – це одна з найрозповсюдженіших технік обробки зображень, яка має на меті покращити якість зображення, завдяки видаленню імпульсних перешкод. Процедури середньоарифметичної фільтрації полягає в тому. що апертура певного обраного розміру проводить скан зображення. Не можна не зазначити, що розмір апертури сильно впливає на ефективність середньоарифметичної фільтрації і обирається з урахуванням розміру зображення та рівня зашумлення зображення імпульсними перешкодами. При цьому серединний елемент апертури замінюється на значення середнього арифметичного точок поточної апертури.

Особливістю і складністю процесу захищеної обробки зображень за допомогою середньоарифметичної фільтрації водночас є те, що під час шифрування значення точки не повинно втрачатись, а лише приховуватись від читання таким чином, щоб результат знаходження середнього арифметичного не змінився і його можна було отримати після здійснення операції дешифрування.

Вимоги, які ставляться до перетворень, що можуть використовуватись для захищеної обробки зображень при середньоарифметичній фільтрації, однозначно забороняють використовувати широкий спектр класів перетворення. Наведені вимоги забороняють використання інтервальних методів шифрування чи використання великого класу нелінійних операцій, особливо операції множення. Результатом застосування якогось з цих перетворень є зміна даних без можливості знаходження зворотнього перетворення для їх відновлення.

Інша, принципово важлива, вимога - це ефективність методів захищеної обробки даних. Разом процедури шифрування і дешифрування мають витрачати менше ресурсів, ніж, безпосередньо, обробка зображення.

Відштовхуючись від цих вимог та особливостей середньоарифметичної обробки зображень, напрямок досліджень - це використання лінійних перетворень, щоб захистити зображення при їхній віддаленій середньоарифметичній фільтрації.

2.1 Організація захищеної групової середньоарифметичної фільтрації зображень з використанням адитивного перемішування

Складність та особливість процесу захищеної обробки зображень за допомогою середньоарифметичної фільтрації полягає у тому, що під час шифрування значення точки не повинно втрачатись, а лише приховатись від прочитання так, щоб результат пошуку середньоарифметичного не змінився та міг бути отриманий після виконання операції дешифрування.

Вимоги, що поставлені до перетворень, котрі можуть використовуватись для захищеного оброблення зображення за допомогою середньоарифметичної фільтрації однозначно забороняють користування широким спектром класів перетворення. Зазначені вище вимоги забороняють використовувати інтервальні методи шифрування або великий клас нелінійних операцій, а особливо операції множення. Як результат застосування будь-якого з наведених перетворень, буде перекручення даних та відсутність знаходження зворотного перетворення для відновлення.

З цих двох вимог і особливостей середньоарифметичної обробки зображень, маємо висновок, ціллю досліджень є використання лінійних перетворень для захисту зображень під час їхньої віддаленої середньоарифметичної фільтрації.

Аналіз можливостей підвищення ефективності захищеної фільтрації зображень в хмарах дозволяє зробити висновок про те, що найбільш перспективним шляхом досягнення поставленої мети є використання перемішування, яке не потребує значних обчислювальних ресурсів. З іншого боку, значний об'єм інформації, що міститься в сучасних зображеннях робить перемішування достатньо ефективним засобом захисту від спроб реконструювання зображень технологіями направленого перебору або статистичного аналізу. Виходячи з цього, в основу розробленого методу захищеної середньоарифметичної фільтрації зображень покладено їх перемішування.

Під час обробки зображення в реальній системі, як правило, необхідне опрацювання не одиночного зображення, а цілого потоку. Це означає, що в процесі розв'язку певної прикладної задачі, система обробляє деяку кількість зображень $\chi = \{A_1, A_2, \ldots, A_n\}$. Ця особливість роботи реальних систем обробки зображень може ефективно використовуватись для підняття ефективності захисту зображень при їхній середньоарифметичної фільтрації на віддалених багатопроцесорних системах.

Як результат проведених досліджень, пропонується використати цю особливість роботи реальних систем обробки зображень як підґрунтя засобу захисту зображень під час їхньої віддаленої середньоарифметичної фільтрації. Запропонований метод захищеної реалізації групової середньоарифметичної фільтрації зображень з використанням адитивного перемішування.

Суть методу захищеної реалізації групової середньоарифметичної фільтрації зображень з використанням адитивного перемішування полягає в тому, щоб здійснити лінійні перетворення над зображеннями A_1, A_2, …, A_q, що складають разом групу з q оригінальних зображень, які потрібно обробити. Як результат лінійних перетворень над зображеннями A_1, A_2, …, A_k, маємо

групу з q захищених зображень G_1, G_2, …, G_q. Кожне з отриманих зашифрованих зображень є матрицею G з s рядків та r стовбців:

$$G = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1r} \\ g_{21} & g_{22} & \cdots & g_{2r} \\ & \cdots & & \\ g_{s1} & g_{s2} & \cdots & g_{sr} \end{bmatrix}.$$

Отриману групу захищених зображень G_1, G_2, …, G_q користувач надсилає на віддалену обчислювальну платформу, на якій здійснюється паралельна обробка цих надісланих зображень методом середньоарифметичної фільтрації. Як результат, G_1, G_2, …, G_q формуються зображення H_1, H_2, …, H_k, що являють собою відфільтровані захищені зображення. Кожне з отриманих зашифрованих зображень є матрицею H з s рядків та r стовбців:

$$H = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1r} \\ h_{21} & h_{22} & \cdots & h_{2r} \\ & \cdots & & \\ h_{s1} & h_{s2} & \cdots & h_{sr} \end{bmatrix}.$$

Віддалена багатопроцесорна система надсилає користувачеві кожне зображення H_u, де $1 \le x \le q$, котре сформоване в результаті середньоарифметичної фільтрації зображення G_x.

Отримавши всю групу з q зображень H_1, H_2, …, H_q, користувач, здійснює їх дешифрування. Для цього він виконує зворотне лінійне перетворення групи зображень H_1, H_2, …, H_q, внаслідок чого він отримує групу з q зображень F_1, F_2, …, F_q . Отриманні в результатті дешифрування зображення F_1, F_2, …, F_q є відфільтрованими відповідниками q оригінальних зображень A_1, A_2, …, A_q.

У методі захищеної реалізації групової середньоарифметичної фільтрації зображень для дешифрування і розшифрування використовуються лінійні перетворення. Для шифрування в запропонованому методі захищеної реалізації групової середньоарифметичної фільтрації зображень пропонується використати систему лінійних перетворень μ, відповідно до якої виконається адитивне лінійне перетасування групи k оригінальних зображень A_1, A_2, …, A_k. Для дешифрування передбачене користування

системою зворотніх лінійних перетворень ρ. Ця система однозначно визначається системою прямих лінійних перетворень μ.

Система лінійних прямих перетворень μ може бути представлена матрицею μ, кожним елементом m_{gh}, де $1 \le g \le q$ та $1 \le l \le q$, якої є коефіцієнт:

$$\mu = \begin{bmatrix} m_{11}\ m_{12} \dots m_{1q} \\ m_{21}\ m_{22} \dots m_{2q} \\ \dots \\ m_{q1}\ m_{q2} \dots m_{qq} \end{bmatrix}.$$

Ссистема зворотніх перетворень ρ, що пов'язана з нею, може бути представлена матрицею μ, кожним елементом p_{gl}, де $1 \le g \le q$ та $1 \le l \le q$, якої є коефіцієнт:

$$\rho = \begin{bmatrix} p_{11}\ p_{12} \cdots p_{1q} \\ p_{21}\ p_{22} \cdots p_{2q} \\ \dots \\ p_{q1}\ p_{q2} \cdots p_{qq} \end{bmatrix}.$$

Користувачеві запропоновано сформувати матриці μ та ρ заздалегідь та використовувати їх як секретні ключі методу групової середньоарифметичної фільтрації зображень з використнням адитивного перемішування.

Користувач завчасно формує певну множину систем перетворень: ω = {< μ_1, ρ_1 >, < μ_2, ρ_2 >, …, < μ_l, ρ_b >}. Тоді, для обробки кожної групи зображень, користувач, випадковим чином, вибирає один секретний ключ з елементів множини ω.

В процесі адитивного перемішування, згідно до правил, визначених обраними перетвореннями μ в рамках поточного секретного ключа з зображень A_1, A_2, …, A_q формуються зображення G_1, G_2, …, G_q. При виконуванні процедури адитивного перемішування, формування зображень G_1, G_2, …, G_q здійснюється обчисленням кожного зображення G_u , значення інтенсивності кольору кожної точки g_{qp} , як композиції точок з відповідними координатами кожного з зображень A_1, A_2, …, A_k.

Якщо представити через a_{uqp} - значення інтенсивності кольору в точці з координатами k, p оригінального зображення A_u, через g_{uqp} - значення інтенсивності кольору в точці з такими ж координатами зображення G_u, то адитивне лінійне перемішування зображень A_1, A_2, …, A_k відповідно правилам, що визначені системою лінійних перетворень μ, може бути представлений, як:

$$g_{ukp} = \sum_{i=1}^{q} m_{ui} \cdot a_{ikp}. \qquad (2.1)$$

Значення вказує на рядок матриці μ, згідно до якого здійснюється адитивне перемішування для утворення точки g_{ukp} зображення G_u.

Провівши аналіз формули (2.1), можна підсумувати, що процес адитивного перемішування виконується незалежно для кожної точки g_{kp} зображення G_u. Окрім цього, незалежно виконується і утворення зображення G_u. Це означає що виконання прооцедури адитивного перемішування може здійснюватись паралельно.

Детальний аналіз свідчить про те, що високу ефективність забезпечує паралельне виконання процедури адитивного перемішування, де одночасно оброблюється q точок всіх зображень G_1, G_2, …, G_q з однаковими координатами k, p.

Зашифровані зображення надсилаються користувачем G_1, G_2, …, G_k на обчислену віддалену платформу та отримує назад відфільтровані зашифровані зображення H_1, H_2, …, H_k.

Для дешифрування відфільтрованих зашифрованих зображень H_1, H_2, …, H_q. користувач здійснює адитивне лінійне перемішування з використуанням системи перетворень ρ у зворотньому напрямку. Якщо представити через h_{ukp} - значення інтенсивності кольору в точці з координатами k, p відфільтрованого зашифрованого зображення H_u, через f_{ukp} - значення інтенсивності кольору в точці з такими ж координатами оригінального відфільтрованого зображення F_u, то процес адитивного зворотнього лінійного перемішування зображень H_1, H_2, …, H_q відповідно

правилам, що визначені системою лінійних перетворень ρ, представлений як:

$$f_{ukp} = \sum_{i=1}^{q} p_{ui} \cdot l_{ikp}. \qquad (2.2)$$

Зробивши детальний аналіз формули (2.2), можна дійти до висновку, що процес дешифрування, при участі адитивного зворотнього перемішування, здійснюється незалежно для кожної точки f_{qp} оригінального відфільтрованого зображення F_u. Кожне зображення F_u теж є незалежно сформованим. Це означає, що процедура дешифрування може виконуватись паралельно.

Проведений аналіз показує, що найвища ефективність досягається, паралельно виконуючи процедуру дешифрування, де одночасно оброблюється q точок всіх зображень H_1, H_2, …, H_q з ідентичними координатами q, p.

Під час виконання середньоарифметичної фільтрації зображень G_1, G_2, …, G_q на багатопроцесорні віддаленій системі, результат обчислень (значення інтенсивностей кольору відфільтрованих зашифрованих зображень H_1, H_2, …, H_q) потрібно зберегти у форматі точки, що плаває, щоб уникнути округлень даних. Це може ввести похибки після дешифрування зображень H_1, H_2, …, H_q.

Фактично, дана процедура захищеної середньоарифметичної групової фільтрації зображень, при участі адитивного перемішування, представлена у вигляді послідовності дій, що зазначена далі:

1) Користувач виділяє з множини χ оригінальних зображень A_1, A_2, …, A_n, що потрібно обробити з k зображень, що оброблюються в межах однієї групи.

2) Користувач випадковим чином обирає систему ортогональних лінійних перетворень μ:

$$\mu = \begin{bmatrix} m_{11} & m_{12} & \dots & m_{1k} \\ m_{21} & m_{22} & \dots & m_{2k} \\ & \dots & & \\ m_{k1} & m_{k2} & \dots & m_{kk} \end{bmatrix}$$

та пов'язану з нею систему ρ зворотних перетворень:

$$\rho = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1k} \\ p_{21} & p_{22} & \cdots & p_{2k} \\ & \dots & & \\ p_{k1} & p_{k2} & \dots & p_{kk} \end{bmatrix}.$$

Системи μ та ρ лінійних перетворень вибираються, опираючись на вимоги, що описані вище.

3) Користувач зашифровує оригінальні зображення A_1, A_2, …, A_k. Для цього користувачем здійснюється адитивне перемішування k зображень поточної групи з допомогою системи μ прямих лінійних перетворень. На вхід системи μ прямих лінійних перетворень подаються оригінальні зображення A_1, A_2, …, A_k.

Як результат адитивного перемішування, користувач має на виході k зашифрованих зображень G_1, G_2, …, G_k.

4) Користувач пересилає сформовані зашифровані зображення G_1, G_2, …, G_k на віддаленні обчислювальні потужності.

5) На обчислювальній віддаленій платформі відбувається обробка кожного зображення G_u за допомогою методу середньоарифметичної фільтрації. В результаті середньоарифметичної фільтрації зашифровних зображень G_1, G_2, …, G_k. отримуються відфільтровані зашифровані зображення H_1, H_2, …, H_k.

$$h_{ij} = \frac{1}{m^2} \sum_{q=i-\frac{m-1}{2}}^{i+\frac{m-1}{2}} \sum_{p=j-\frac{m-1}{2}}^{j+\frac{m-1}{2}} g_{qp}. \qquad (2.3)$$

При цьому, обробка середньоарифметичною фільтрацією кожного зображення G_u виконується незалежно і одночасно залучає k процесорів системи.

6) В результаті обробки методом середньоарифметичної фільтрації захищених зображень G_1, G_2, …, G_k, сформовані захищенні відфільтровані

зображення H_1, H_2, …, H_k передаються користувачеві з віддаленої обчислювальної платформи.

7) Після отримання відфільтрованих захищених зображень H_1, H_2, …, H_k користувач їх дешифрує. Щоб зробити це, він використовує систему ρ зворотніх лінійних перетворень, на вхід якої подає зображення H_1, H_2, …, H_k. Результат дешифрування - отримання відфільтрованих оригінальних зображень F_1, F_2, …, F_k.

Критеріями оцінки ефективності методу захищеної реалізації групової середньоарифметичної фільтрації зображень з використанням адитивного перемішування на обчислювальних віддалених платформах, варто поставити:

-швидкодію;

-криптостійкість.

Оцінку ефективності методу захищеної реалізації групової середньоарифметичної фільтрації зображень з використанням адитивного перемішування на віддалених обчислювальних платформах можна поставити наступним чином. Під оцінкою швидкодії в даному випадку підрозумовується оцінювання пришвидшення виконання середньоарифметичної фільтрації з залученням віддалених обчислювальних потужностей в порівнянні з середньоарифметичною фільтрацію, використовуючи виключно ресурси користувача.

Операції шифрування і дешифрування можна звести до виконання адитивного перемішування в межах групи з k зображень. Адитивне перемішування передбачає формування k зображень у результаті виконання k лінійних перетворень, на вхід кожного з яких подається k вхідних зображень (в межах однієї групи).

Загальний час виконання шифрувння і дешифрування однієї групи зображень на одному процесорі з використанням адитивного перемішування визначається наступною формулою:

$$t_{cdmix} = 2 \cdot k^2 \cdot s \cdot r \cdot t_s. \quad (2.4)$$

В реальних системах, котрим для розв'язку практичних задач необхідна середньоарифметична фільтрація зображень, як правило, виконується обробка не одиночних зображень, а їхнього потоку. Іншими словами, в той час, коли на віддаленій платформі виконується середньоарифметична фільтрація поточної групи з k зображень, користувачем може бути проведене шифрування наступної групи з k оригінальних зображень. Таким чином, час, що витрачається користувачем для обробки однієї групи зображень, згідно з методом захищеної реалізації групової середньоарифметичної фільтрації зображень, визначається лише часом шифрування і дешифрування однієї грипи зображень.

Пришвидшення здійснення середньоарифметичної фільтрації із участю віддалених обчислювальних потужностей, порівнюючи з середньоарифметичною фільтрацією, використовуючи лише ресурси користувача, виражається через коефіцієнт σ. Коефіцієнт прискорення σ визначається співвідношенням часу t_f фільтрації зображення на обчислювальний платформі користувача до часу t_{cdmix} обчислень, що пов'язані із шифруванням зображення перед тим, як надіслати його для обробки віддалено та дешифруванням його після повернення до користувача, методом адитивного перемішування:

$$\sigma = \frac{t_f}{t_{cdmix}} = \frac{k \cdot s \cdot r \cdot (m^2 \cdot t_s + t_d)}{2 \cdot k^2 \cdot s \cdot r \cdot t_s} = \frac{m^2}{2 \cdot k} + \frac{t_d}{2 \cdot k \cdot t_s}. \quad (2.5)$$

Відповідно до оцінок [14], час t_d виконання однієї операції арифметичного ділення сучасними процесорами Intel є в 30 раз більшим, порівнюючи з часом t_s виконання однієї операції арифметичного додавання. Врахувавши це, коефіцієнт прискорення σ представлений спрощено. А саме:

$$\sigma = \frac{m^2 + 30}{2 \cdot k}. \quad (2.6)$$

При типовому для практичних задач розмірі апертури, m = 13, коефіцієнт σ прискорення становить:

$$\sigma = \frac{100}{\kappa}$$

Якість захищеності зображень, що передаються потенційно відкритими каналами мережі і оброблюються на віддалених, потенційно відкритих, багатопроцесорних системах, які не контролюються користувачем, може оцінюватись через об'єм ресурсів, які зловмисник має витратити для того, щоб отримати оригінальні зображення A_1, A_2, …, A_k або їхніх відфільтрованих відповідників F_1, F_2, …, F_k із зашифрованих зображень G_1, G_2, …, G_k .

Щоб провести оцінку захищеності зображень вважатимемо, що зловмисник має засоби ідентифікації, щоб довести оригінальність зображення. Тобто під час взлому, зловмисник чітко зрозуміє яке саме перед ним зображення – дійсне оригінальне чи випадкове інше. Він може визначити це за допомогою певних значень статистичних показників, які характерні для цього класу зображень [20].

Вище було зазначено, що секретним ключем в запропонованому методі захищеної реалізації групової середньоарифметичної фільтрації зображень є системи лінійних ортогональних перетворень μ та ρ. Для відновлення оригінальних зображень A_1, A_2, …, A_k чи їхніх відфільтрованих відповідників F_1, F_2, …, F_k з зашифрованих зображень G_1, G_2, …, G_k чи їхніх відфільтрованих відповідників F_1, F_2, …, F_k , зловмиснику треба підібрати систему μ лінійних ортогональних перетворень або систему ρ лінійних ортогональних перетворень, що є їй зворотньою.

Оцінкою об'єму ресурсів, що має витратити зловмисник для отримання оригінальних зображень A_1, A_2, …, A_k чи їхніх відфільтрованих відповідників F_1, F_2, …, F_k з зашифрованих зображень G_1, G_2, …, G_k чи їхніх відфільтрованих відповідників F_1, F_2, …, F_k , може виступати загальна кількість Q(k) систем μ лінійних ортогональних перетворень.

Варто також звернути увагу на те, що якщо елементи матриць μ та ρ належать множині значень {-1, 0, 1}, то процес шифрування і

дешифрування є найбільш швидким, тому що потребує виконання виключно операцій арифметичного додавання чи віднімання.

Обчислення кількості Q(k) систем μ лінійних ортогональних перетворень, що залежить від значення кількості зображень в одній групі, може застосувати комбінаторні методи. Згідно ортогональним лінійним системам в алгебрі Жегалкіна, загальна кількість лінійних булевих функцій від k змінних дорівнює 2^k-1. Слід згадати, що значення кількості лінійних булевих функцій від k змінних показане без врахованих інверсій цих функцій.

Перша функція системи μ лінійних ортогональних перетворень може бути обрана 2^k-1 способами. Специфіка задач захисту даних не дає використовувати булеві функцій, де функція співпадає зі змінними. В матриці μ не може бути такого рядка, котрий містив би тільки один елемент, що дорівнює одиниці. У протилежному випадку, це означає, що одне з зашифрованих зображень G_1, G_2, …, G_k співпадає з одним із оригінальних A_1, A_2, …, A_k. Дана ситуація означає, що зображення з множини {A_1, A_2, …, A_k} передається відкритими каналами на віддалену і також відкриту систему в незахищеному вигляді. Це протирічна ситуація з основною умовою задачі віддаленої захищеної обробки.

Специфіка задачі із захистом даних, при їхній обробці віддалено на відкритих системах, змушує знизити кількість можливих лінійних булевих функцій, що можуть бути використанні на величину k. Відповідно до цього, перша функція системи μ лінійних ортогональних перетворень може бути обрана 2^k-1-k способами. Тобто першою функцією може бути будь-яка з лінійних булевих функцій алгебри Жегалкіна, крім тих що співпадають зі змінними.

Друга функція системи μ лінійних ортогональних перетворень може бути обрана 2^k-1-k способами, при умові, що друга функція не повинна співпадати з першою. Дана умова необхідна для того, щоб система μ

лінійних перетворень була ортогональною. Іншими словами, щоб сума або різниця будь-якої підмножини рядків матриці μ не дорівнювала нулю. Виходячи з цього, друга функція системи μ лінійних ортогональних перетворень обирається 2^k-2-k способами.

Третя функція може бути обрана 2^k-1-k способами, але при умові, що друга функція не співпадає з першою чи з другою. Крім цього, для виконання умови, що сума чи різниця будь-якої підмножини рядків матриці μ не рівна нулю, третя функція не повинна співпадати з лінійною комбінацією першої та другої функцій. Як наслідок, маємо третю функцію системи μ лінійних ортогональних перетворень, котра може бути обрана лише 2^k-4-k способами. Під час вибору кожної наступної функції системи μ лінійних ортогональних перетворень також потрібно зважати умови, описані вище. Для будь-якої u-тої функції системи μ лінійних ортогональних перетворень, число функції, з яким вона не повинна співпадати дорівнює числу лінійних комбінацій u-1 попередньо обраних функцій. При виборі u-тої функції системи μ лінійних ортогональних перетворень виключається 2^{u-1}-1 функцій з можливого набору лінійних булевих функцій алгебри Жегалкіна. Це означає, що будь-якою u-тою функцією системи μ лінійних ортогональних перетворень може бути обрана з 2^k-2^{u-1}-1-k функцій.

Маємо, що загальна кількість Q(k) можливих систем адитивного лінійного перемішування оригінальних зображень, що представлена кількістю секретних ключів запропонованого методу захищеної реалізації середньоарифметичної фільтрації зображень, з використанням методу адитивного маскування, може бути обчислена як поліном k кількостей можливих для вибору функцій, при виборі кожної u-тої функції:

$$Q(k) = \prod_{u=1}^{k}(2^k - \eta - 2^{u-1}). \qquad (2.7)$$

При кількості зображень в одній групі k = 8, кількість секретних ключів $Q(k) = 4 \cdot 10^{18}$. Оскільки для доступу до оригінальних зображень A_1, A_2, …, A_k чи їх відфільтрованих відповідників F_1, F_2, …, F_k з зашифрованих зображень

G_1, G_2, …, G_k чи їхніх відфільтрованих відповідників F_1, F_2, …, F_k зловмисник має знати системи μ лінійних ортогональних перетворень або систему ρ лінійних ортогональних перетворень, зворотну до неї, то йому потрібно перебрати близько $2 \cdot 10^{18}$ варіантів.

Оскільки кожна група зашифрована через інший ключ, який користувач обирає випадковим чином з множини сформованих до цього ключів, при знаходженні системи μ лінійних ортогональних перетворень або зворотної до неї системи ρ лінійних ортогональних перетворень, зловмисник отримає доступ лише до поточної групи зображень. Вже навіть при $k = 8$, ідея виконання перебору секретних ключів є не практично та недоцільною.

Відповідно до запропонованого методу захищеної реалізації середньоарифметичної фільтрації зображень з використанням методу адитивного маскування, передбачена процедура шифрування через лінійне перемішування зображень є ефективною при кількості зображень однієї групи k = 6…8. Вже при значеннях k в межах цього діапазону, даний метод захищеної реалізації середньоарифметичної фільтрації зображень з використанням адитивного маскування успішно протидіятиме статистичному відновленню зображень у випадку їх перехоплення зловмисниками.

Для більшої гарантії рівня захищеності зображень, рекомендується обирати їх різні класи, під час передачі каналами, що є потенційно відкритими та обробки на потенційно відкритих віддалених платформах, для обробки в межах однієї групи. Зокрема, при обробці методом захищеної реалізації середньоарифметичної фільтрації зображень, не буде доцільно формувати групу зображень, котрі є суміжними за часом своєї зйомки.

Отже, метод групової захищеної реалізації середньоарифметичної фільтрації зображень з використанням методу адитивного маскування може забезпечити високий рівень захисту зображень від спроб зловмисників несанкціоновано отримати оригінальні зображення A_1, A_2, …, A_k чи їхніх

відфільтровані відповідники F_1, F_2, …, F_k з зашифрованих зображень G_1, G_2, …, G_k чи їхніх відфільтрованих відповідників F_1, F_2, …, F_k, під час їх передачі каналами мережі, котрі є потенційно відкритими та під час обробки на віддалених багатопроцесорних системах.

Важлива особливість методу групової захищеної реалізації середньоарифметичної фільтрації зображень - це можливість гнучкої зміни рівня захищеності зображень, в залежності від вимог користувача до криптостійкості. Це є прикладом яскравої переваги запропонованого методу, адже вона дозволяє користувачу знайти компроміс між рівнем захищеності та швидкістю обробки зображень при використанні цього методу. Також, збільшення k зображень в одній групі підвищує рівень захищеності, хоча має наслідком збільшення часу виконання процедур шифрування на дешифрування зображень, а це є визначальним для користувача.

2.2 Захищена організація групої середньоарифметичної фільтрації зображень з використнням адитивного маскування і перемішування

Основа середньоарифметичної фільтрації зображень - це виконання операції обчислення середньоарифметичного точок зображення в межах поточної апертури. Це операція, котра складається з арифметичного додавання та ділення. Найпростіший та оптимальний підхід до шифрування операндів арифметичного додавання - це їхнє адитивне маскування.

В рамках цього підходу, застосовується метод організації захищеної реалізації середньоарифметичної фільтрації зображень, на віддалених обчислювальних платформах, що можуть бути потенційно відкриті, з використанням адитивного маскування. Цей метод підрозумовує накладання точок випадкового маскуючого зображення на відповідні точки оригінального зображення.

Особливістю та найбільшою перевагою цієї процедури є висока ефективність, стосовно оцінювання критерію швидкодії. Фактично, операції шифрування та дешифрування при адитивному маскуванні вимагають виконання лише однієї операції додавання чи віднімання тільки для одного елементу зображення.

Для реальних систем, яким для вирішення практичних задач потрібно виконати середньоарифметичної фільтрації, як правило, здійснюється обробка не одиночних зображень, а їх цілого потоку. Тому, під час віддаленої середньоарифметичної фільтрації поточного зображення, користувач може виконувати шифрування наступного. Відповідно, час, який користувач використовує безпосередньо, щоб обробити кожне зображення згідно з запропонованим методом, визначається часом його шифрування і дешифрування.

Але слід зазначити, що просте адитивне маскування має два недоліки:

1) Зображення-маска передається по відкритим каналам Інтернет і оброблюється на непідконторольних обчислювальних потужностях. Так, існує потенційна небезпека співставлення захищеного зображення та його маски.

2) Якщо маска незалежна *M* від оригінального зображення, елементи якої формуються випадково, існує реальна загроза часткового відновлення оригінального зображення *A* з замаскованого *G* статистичними методами. Це вказує на обмеження фільтрування масок у відкритому вигляді на віддаленних обчислювальних потужностях.

Операції шифрування та дешифрування зводяться до виконання адитивного перемішування в межах грузи з k зображень. Адитивне перемішування, в межах однієї групи, передбачає формування k зображень внаслідок виконання k лінійних перетворень, на вхід кожного з яких подається k вхідних зображень.

Для реальних систем, яким для вирішення практичних задач потрібно виконати середньоарифметичної фільтрації, як правило, здійснюється обробка не одиночних зображень, а їх цілого потоку. Тому, під час віддаленої середньоарифметичної фільтрації поточного зображення, користувач може виконувати шифрування наступного. Відповідно, час, який користувач використовує безпосередньо, щоб обробити кожне зображення згідно з запропонованим методом, визначається часом його шифрування і дешифрування. Зокрема, при кількості зображень в одній групі k = 8, кількість секретних ключів $Q(k) = 4 \cdot 10^{18}$. Оскільки для доступу до оригінальних зображень A_1, A_2, …, A_k чи їхніх відфільтрованих відповідників F_1, F_2, …, F_k з зашифрованих зображень G_1, G_2, …, G_k чи їхніх відфільтрованих відповідників F_1, F_2, …, F_k зловмиснику мають бути відомі системи μ лінійних ортогональних перетворень або зворотна до неї система ρ лінійних ортогональних перетворень, то виходить, що йому потрібно перебрати близько $2 \cdot 10^{18}$ варіантів.

Якщо кожна група шифрується індивідуальним ключем, що обирається випадково користувачем з множини сформованих раніше ключів, то при знаходженні системи μ лінійних ортогональних перетворень чи зворотної до неї системи ρ лінійних ортогональних перетворень, зловмисник зможе отримати доступ лише до поточної групи зображень. Вже навіть при k = 8, ідея виконання перебору секретних ключів є практично недоцільною.

Згідно з методом захищеної реалізації середньоарифметичної фільтрації зображень з використанням методу адитивного маскування, передбачена процедура шифрування, при участі лінійного перемішування зображень є ефективною при кількості зображень однієї групи k = 6…8. Вже при значеннях k в межах цього діапазону запропонований метод може успішно протидіяти спробам статистичного відновлення зображень, якщо зловмисники їх перехоплять.

Детальний аналіз таблиці 2.1 свідчить про те, що при збільшенні значення кількості k зображень в одній групі, кількість $Q(k)$ секретних ключів стрімко збільшується. Зокрема, при кількості зображень в одній групі k = 8, кількість секретних ключів $Q(k) = 4 \cdot 10^{18}$.

Для більшої гарантії рівня захищеності зображень, при їхній передачі каналами, що є потенційно відкритими та обробки на, також потенційно відкритих, віддалених платформах, для обробки в межах однієї групи, рекомендовано вибирати зображення різних класів. Зокрема, при обробці з використанням запропонованого методу, не рекомендується формувати групу з зображень, що суміжні за часом зйомки.

Другий недолік надзвичайно важливий лише для обмеженого класу зображень. Проведений аналіз показує, що статистичні методи дозволяють отримати доступ лише до окремих контурів зображень. Якщо говорити інакше, то цей недолік дуже небезпечний лише для класу зображень, у яких основною інформаційною складовою є чітко окресленні контури. Цей клас включає значну частину аерокосмічних зображень.

Звісна річ, існує клас прикладних задач, для яких наближеного відновлення даних не достатньо для доступу до інформації. Це відноситься до тих знімків, де дрібні деталі містять основну інформацію, а не загальні контури. Зокрема, до них відносяться фото людей, які використовуються для їх ідентифікації, знімки для дистанційного слідкування за рухами обраних об’єктів. Але треба зазначити, що другий недолік вносить значні обмеження на використання адитивного маскування, як елементу захисту при захищеній віддаленій середньоарифметичній фільтрації.

З урахуванням переваг і недоліків, що перелічені та зазначені вище, методів захищеної реалізації середньоарифметичної фільтрації зображень, пропонується метод захищеної реалізації групової середньоарифметичної фільтрації зображень з використанням адитивного маскування та адитивного перемішування. Це метод, який може передбачити виконання

простого адитивного маскування оригінальних зображень та адитивного перемішування замаскованих оригінальних зображень в межах визначеної групи.

Методом захищеної реалізації групової середньоарифметичної фільтрації зображень передбачається, те що формування набору з z масок B_1, B_2, …, B_z здійснюється користувачем заздалегідь та наперед. Кожна, з сформованого набору z масок, маска B являється матрицею з s рядків та r стовбців, елементами якої є випадкові значення:

$$B = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1r} \\ b_{21} & b_{22} & \dots & b_{2r} \\ & \dots & & \\ b_{s1} & b_{s2} & \dots & b_{sr} \end{bmatrix}.$$

Використовуючи свої власні обчислювальні ресурси, користувачем наперед виконується середньоарифметична фільтрація z масок B_1, B_2, …, B_z з використанням апертури розміру m . Як результат, користувач отримує z зображень E_1, E_2, …, E_z, котрі є відфільтрованими відповідниками попередньо сформованих масок B_1, B_2, …, B_z:

$$E = \begin{bmatrix} e_{11} & e_{12} & \dots & e_{1r} \\ e_{21} & e_{22} & \dots & e_{2r} \\ & \dots & & \\ e_{s1} & e_{s2} & \dots & e_{sr} \end{bmatrix}.$$

Тобто зображення E формується як:

$$e_{ij} = \frac{1}{m^2} \sum_{q=i-\frac{r-1}{2}}^{i+\frac{r-1}{2}} \sum_{p=j-\frac{r-1}{2}}^{j+\frac{r-1}{2}} a_{qp}. \qquad (2.8)$$

З набору з z масок B_1, B_2, …, B_z , що вже є сформованим, користувач обирає η масок B_1, B_2, …, B_η, які використовуватимуться для маскування η оригінальних зображень A_1, A_2, …, A_η. При чому, значення величини η в декілька разів менше за кількість z масок, сформованих попередньо, тобто $\eta << z$.

Позначивши через C_u замасковане, за допомогою маски B_u, зображення, воно є матрицею з s рядків та r стовбців:

$$C = \begin{bmatrix} c_{11}\ c_{12} \dots c_{1r} \\ c_{21}\ c_{22} \dots c_{2r} \\ \dots \\ c_{s1}\ c_{s2} \dots c_{sr} \end{bmatrix}.$$

Зображення C запропоновано отримати, використовуючи шлях адитивного маскування точок зображення A точками зображення B. Формально процес шифрування оригінального зображення A зводиться до додавання матриць A та M:

$$C = A + B.$$

і здійснює операції передбаченні простим адитивним маскуванням відносно w оригінальних зображень A_1, A_2, …, A_w, з отриманням G_1, G_2, …, G_w. Наступним етапом є адитивне перемішування w замаскованих зображень G_1, G_2, …, G_w.

Суть адитивного перемішування полягає в тому, що над групою з w зображень G_1, G_2, …, G_w здійснюються лінійні перетворення, з отриманням w зображень V_1, V_2, …, V_w, які будуть оброблятись на віддалених багатопроцесорних системах.

В такому разі, захищена групова середньоарифметична фільтрація зображень складається з наступної послідовності дій:

1)Користувач обирає w зображень A_1, A_2, …, A_w котрі фільтруватимуться в межах однієї групи.

2)Користувач обирає з попередньо створених q масок, w масок M_1, M_2, …, M_w.

3)Кожне зображення A_p, $p \in \{1, 2, \dots, w\}$, шифрується маскою M_p за допомогою адитивного маскування, з утворенням w замаскованих зображень G_1, G_2, …, G_w.

4)Користувач адитивно перемішує за допомогою системи перетворень Λ, замаскованні зображення G_1, G_2, …, G_w і отримує захищенні зображення зображень V_1, V_2, …, V_w.

5)Користувач надсилає захищенні зображення V_1, V_2, …, V_w на обчислювальні віддалені потужності.

5)На віддалених багатопроцесорних системам кожне зображення V_p незалежно оброблюється за допомогою середньоафрифметичної фільтрації, з отриманням зображення Q_p.

6)Відфільтроване захищене зображення Q_p передається користувачеві.

7)Користувач, отримавши всю групу зображень Q_1, Q_2, …, Q_w виконує зворотні перетворення системи Г з отриманням w замаскованих відфільтрованих зображень U_1, U_2, …, U_w.

12) Користувач відновлює відфільтрованні замасковані зображення U_1, U_2, …, U_w, за допомогою D_1, D_2, …, D_w, з отриманням відфільтрованих оригінальних зображень S_1, S_2, …, S_w.

Описана процедура захисту зображень в процесі його середньоарифметичної фільтрації може бути ілюстрована наступним прикладом:

Сформоване за описаною процедурою зображення Q являє собою відфільтроване оригінальне зображення P.

Запропонований метод захищеної віддаленої середньоарифметичної фільтрації зображень ілюструється наступним прикладом.

Нехай оригінальне зображення P з 6-ти рядків ($k = 6$) і 6-ти стовбців ($h = 6$) має наступний вигляд:

$$P = \begin{pmatrix} 1 & 1 & 1 & 2 & 4 & 6 \\ 1 & 1 & 1 & 2 & 5 & 9 \\ 1 & 5 & 1 & 3 & 1 & 7 \\ 1 & 1 & 2 & 3 & 1 & 8 \\ 1 & 2 & 3 & 4 & 3 & 8 \\ 2 & 2 & 3 & 4 & 5 & 9 \end{pmatrix}.$$

В результаті середньоарифметичної фільтрації цього зображення апертурою 3 х 3 ($a = 3$) формується наступна матриця Q відфільтрованого зображення:

$$Q = \begin{pmatrix} 1 & 1 & 1 & 2 & 4 & 6 \\ 1 & 1{,}44444 & 1{,}88889 & 2{,}22222 & 4{,}33333 & 9 \\ 1 & 1{,}55556 & 2{,}11111 & 2{,}11111 & 4{,}33333 & 7 \\ 1 & 1{,}88889 & 2{,}66667 & 2{,}33333 & 4{,}22222 & 8 \\ 1 & 1{,}88889 & 2{,}66667 & 3{,}11111 & 5 & 8 \\ 2 & 2 & 3 & 4 & 5 & 9 \end{pmatrix}$$

Після округлення значень елементів матриці , користувач отримує матрицю *Q* відфільтрованого оригінального зображення:

$$Q = \begin{pmatrix} 1 & 1 & 1 & 2 & 4 & 6 \\ 1 & 1 & 2 & 2 & 4 & 9 \\ 1 & 2 & 2 & 2 & 4 & 7 \\ 1 & 2 & 3 & 2 & 4 & 8 \\ 1 & 2 & 3 & 3 & 5 & 8 \\ 2 & 2 & 3 & 4 & 5 & 9 \end{pmatrix}.$$

З використанням таблиці *TD* користувач формує перемішане зображення *G*, матриця якого має наступний вигляд:

$$G = \begin{pmatrix} 2 & 6 & 1 & 4 & 1 & 1 \\ 2 & 9 & 1 & 5 & 1 & 1 \\ 3 & 7 & 1 & 1 & 1 & 5 \\ 3 & 8 & 1 & 1 & 2 & 1 \\ 4 & 8 & 1 & 3 & 3 & 2 \\ 4 & 9 & 2 & 5 & 3 & 2 \end{pmatrix}.$$

Ця матриця надсилається користувачем на хмару. На віддаленій комп'ютерній системі виконується часткова середньоарифметична фільтрація перемішаного зображення *G* з формуванням матриці *R*:

$$R = \begin{pmatrix} 2 & 6 & 1 & 4 & 1 & 1 \\ 0{,}777778 & 2{,}44444 & 0{,}333333 & 1{,}11111 & 0{,}333333 & 0{,}777778 \\ 0{,}888889 & 2{,}66667 & 0{,}333333 & 0{,}777778 & 0{,}444444 & 0{,}777778 \\ 1{,}11111 & 2{,}55556 & 0{,}333333 & 0{,}555556 & 0{,}666667 & 0{,}888889 \\ 1{,}22222 & 2{,}77778 & 0{,}444444 & 1 & 0{,}888889 & 0{,}555556 \\ 4 & 9 & 2 & 5 & 3 & 2 \end{pmatrix}.$$

Це зображення пересилається користувачеві.

Користувач здійснює копіювання крайніх точок оригінального зображення *P* в матрицю зображення *Q*. Значення елементів другого стовбця (з другого по п'ятий в кожному стовбці матриці *Q* обчислюється за формулою (10), а

всіх інших – за формулою (11). В результаті користувачем отримується наступна матриця Q:

$$Q = \begin{pmatrix} 1 & 1 & 1 & 2 & 4 & 6 \\ 1 & 1{,}44444 & 1{,}88889 & 2{,}22222 & 4{,}33333 & 9 \\ 1 & 1{,}55556 & 2{,}11111 & 2{,}11111 & 4{,}33333 & 7 \\ 1 & 1{,}88889 & 2{,}66667 & 2{,}33333 & 4{,}22222 & 8 \\ 1 & 1{,}88889 & 2{,}66667 & 3{,}11111 & 5 & 8 \\ 2 & 2 & 3 & 4 & 5 & 9 \end{pmatrix}.$$

.

Виконавши округлення елементів матриці, користувач отримує матрицю Q відфільтрованого оригінального зображення:

$$Q = \begin{pmatrix} 1 & 1 & 1 & 2 & 4 & 6 \\ 1 & 1 & 2 & 2 & 4 & 9 \\ 1 & 2 & 2 & 2 & 4 & 7 \\ 1 & 2 & 3 & 2 & 4 & 8 \\ 1 & 2 & 3 & 3 & 4 & 8 \\ 2 & 2 & 3 & 4 & 5 & 9 \end{pmatrix}.$$

Доцільно здійснювати оцінку ефективності запропонованого методу захищеної групової середньоарифметичної фільтрації зображень за двома критеріями:

- прискоренням, що забезпечує запропонований метод, в порівнянні із виконанням середньоарифметичної фільтрації зображень на обчислювальних ресурсах користувача;
- надійністю, стосовно забезпечення конфіденційності особистих оброблюваних зображень.

Секретний ключ даного методу - маски M_1, M_2, …, M_w та системи Λ і Γ ортогональних лінійних перетворень. Щоб повністю відновити оригінальні зображення, зловмисник повинен дізнатись систему перетворення Λ або зворотньою до неї маски Γ та маски M_1, M_2, …, M_w або їх відфільтровані відповідники D_1, D_2, …, D_w. Оскільки в запропонованому методі використовується одна з можливих систем Λ та w з q масок в довільній комбінації, то час взлому запропонованого методу захисту можна оцінити

об’ємом ресурсів для перебору всіх можливих комбінацій масок та систем лінійних перетворень.

В якості оцінки ресурсів, що має витратити зловмисник для відновлення вибраної користувачем пари систем Λ, може слугувати загальна кількість $N(w)$ таких систем. Якщо вважати, що для всіх матриць Λ $\lambda_{ij} \in \{-1,\ 0,\ 1\}$, то кількість $N(w)$ можливих ортогональних систем рівна кількості ортогональних лінійних систем в алгебрі Жегалкіна [18]. Підрахунок кількості $N(w)$ в залежності від кількості w змінних можна виконати комбінаторними методами. Загальна кількість лінійних булевих функцій від w змінних дорівнює 2^w-1(без урахування їх інверсій). Відповідно, перша функція системи Λ може бути вибрана 2^w-1-w способами, тобто на роль першої функцій системи Λ підходить будь-яка функція з 2^w-1 крім тих, що співпадають зі змінними. Якщо в матриці Λ існує рядок з однією одиничною компонентою, це означає, що одне з зашифрованих зображень G_1, G_2, …G_η співпадає з оригінальним, тобто одне зображення передається в відкритому вигляді. Оскільки умови задачі захисту виключають можливість передачі зображення в явному вигляді, то кількість можливих лінійних булевих функцій, що можуть використовуватися для адитивного лінійного перемішування, зменшується на w.

Друга функція системи Λ може бути вибрана з 2^w-1-w так, щоб вона не співпалдала з першою, тобто 2^w-2-w способами. Третя функція системи Λ може бути вибрана з 2^w-1-w так, щоб вона не співпалдала з двома раніше обраними функціями та їх лінійною комбінацією, тобто з 2^w-4-w. Аналогічно z-та, де $z \in \{1,\ 2,\ \ldots, w\}$ функція системи Λ може бути вибрана з 2^w-4-w так, щоб не співпадала з будь-якою 2^{z-1}-1 лінійних комбінацій раніше обраних z-1 функцій, тобто число варіантів вибору становить 2^w-2^{z-1}-w.

Таким чином загальна кількість $N(w)$ систем Λ обчислюється як :

$$N(w) = \prod_{z=1}^{w}(2^w - w - 2^{z-1}). \qquad (2.8)$$

c_M - вартість перебору всіх можливих поєднань (комбінацій) масок

В такому разі, час адитивного маскування та адитивного перемішування групи зображень рівний часу T_{1w} фільтрації однієї групи з w зображень сумарному:

$$T_{1w} = w \cdot t_{c-d} + t_{mix},$$

де t_{mix} - час виконання перетворень Λ і Γ в рамках однієї групи w масок.

При умові, що $\lambda_{ij} \in \{-1,\ 0,\ 1\}$, w лінійних перетворень системи Λ включають лише операції додавання або віднімання відповідних елементів w зображень G_1, G_2, …, G_w. Оскільки перетворення системи Γ, зворотної до Λ, теж є лінійними, то час їхнього виконання є аналогічним. (Оскільки Γ є системою зворотних лінійних перетворень відносно системи Λ, то не існує різниці в часі їхнього виконання.) Таким чином час t_{mix} рівний:

$$t_{mix} = 2 \cdot w^2 \cdot k \cdot h \cdot t_a. \qquad (2.9)$$

Загальний час T_{1w} фільтрації однієї групи з w зображень становить:

$$T_{1w} = w \cdot k \cdot h(2 \cdot t_a + w \cdot t_a). \qquad (2.10)$$

Відповідно прискорення ϕ, яке забезпечується за умови захищеної віддаленої середньоарифметичної фільтрації з адитивними маскуванням та перемішуванням може бути обчислене як:

$$\phi = \frac{w \cdot k \cdot h \cdot ((r+2) \cdot t_a + t_d)}{w \cdot k \cdot h(2 \cdot t_a + w \cdot t_a)} = \frac{(r+2) \cdot t_a + t_d}{2 \cdot t_a + w \cdot t_a} \approx \frac{(r+2)+30}{2+w}. \qquad (2.11)$$

Для реальних зображень найменша кількість h стовбців складає 1024, відповідно, кількість η варіантів перестановок стовбців становить $6 \cdot 10^{2639}$. Це практично унеможливлює відновлення оригінального зображення шляхом підбору зворотної перестановки, оскільки перебір такої кількості варіантів виходить за рамки можливостей технічної реалізації.

Для певних класів контурних зображень об'єм перебору може бути суттєвим чином зменшено за рахунок направленої реконструкції зображення. Ця технологія передбачає вибір стовбців таким чином, щоб два сусідніх мінімально відрізнялись один від одного. Проведені експериментальні дослідження показали, що для реальних контурних зображень об'єм перебору може бути зменшено на 2-3 порядки. При

захищеній обробці зображень цих класів з використанням розробленого методу рекомендується організовувати одночасну фільтрацію групи з n зображень. При цьому стовпці n обраних зображень перемішуються в межах групи. Кількість зображень в межах однієї групи не впливає на час виконання шифрування зображення перед передачею його в мережу та завершальну фазу фільтрації. Проте кількість варіантів перестановок збільшується до $(n \cdot h)!$, що суттєво підвищує рівень захищеності зображень.

Розроблений метод захищеної фільтрації на основі перемішування стовпців дозволяє, за рахунок використання віддалених обчислювальних потужностей, прискорити цю операцію на 1-2 порядки, що практично збігається з аналогічними показником найбільш швидкодіючого варіанту захисту зображень на основі адитивного маскування.

Основна перевага розробленого методу полягає в більш високому рівні захищеності від спроб, з використанням статистичного аналізу, отримати незаконний доступ до зображень під час їх обробки на непідконтрольних користувачу віддалених комп'ютерних системах.

Запропонований метод орієнтований на широке коло застосувань, пов'язаних з аналізом потоків зображень для підвищення оперативності їх обробки за рахунок використання можливостей сучасних хмарних технологій.

ВИСНОВКИ ДО РОЗДІЛУ 2

В результаті проведених теоретичних розробок досліджень, які складають другий розділ дипломного проекту, і мають на меті пришвидшення середньоарифметичної фільтрації зображень за рахунок використання високопродуктивних віддалених обчислювальних потужностей, можна зробити такі висновки:

1. Аналіз практики обробки аерокосмічних зображень доводить, що в більшості випадків відбувається фільтрація не одиночного зображення, а обробка потоку зображень. Це може бути використано для підвищення ефективності віддаленої захищеної фільтрації зображень в хмарах.

2. Досліджено теоретично метод групової захищеної фільтрації, в якому функції захисту реалізуються за рахунок контрольованого перемішування зображень групи. В якості секретного ключа для дешифрування виступає таблиця перемішування, яка генерується користувачем і зберігається в секреті. Показано, що підбір перестановочної таблиці на віддалених комп'ютерних системах виходить за рамки технічних можливостей сучасних комп'ютерних систем.

3. Запропонована модифікація методу захищеної віддаленої середньоарифметичної фільтрації, яка поєднує переваги групової обробки зображень з їх перемішуванням та адитивного маскування. Використання модифікованого алгоритму віддаленої середньоарифметичної фільтрації дозволяє отримати більший ефект прискорення віддаленої реалізації фільтрації.

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМ ДЛЯ МОДЕЛЮВАННЯ ВІДДАЛЕНОЇ ЗАХИЩЕНОЇ СЕРЕДНЬОАРИФМЕТИЧНОЇ ФІЛЬТРАЦІЇ ЗОБРАЖЕНЬ

Для експериментального дослідження викладеної в другому розділі дипломного проекту методу середньоарифметичної фільтрації зображень з використанням віддалених комп'ютерних систем в рамках виконання дипломного проекту розроблені спеціальні програмні засоби.

Основна мета розробка – створити засоби для:

- функціональної перевірки коректності роботи методу захищеної середньоарифметичної фільтрації зображень, в тому числі і в критичних режимах його роботи;
- статистичного аналізу зашифрованих зображень для реалістичної оцінки загрози відповлення зловмисником зашифрованого зображення шляхом статистичної контуріалізації;
- експериментальної оцінки часу фільтрації на віддалених комп'ютерних системах, виконання операцій шифрування та дешифрування на обчислювальній платформі користувача, а також експериментальної оцінки коефіцієнту прискорення середньоарифметичної фільтрації за рахунок використання віддалених високопродуктивних комп'ютерних систем.

Для досягнення вказаних цілей в рамках дипломного проекту розроблено програму моделювання віддаленої захищеної середньоарифметичної фільтрації зображень.

Оскільки однією з цілей розробленої програми є функціональне моделювання, то програма дозволяє проводити роботи методу віддаленої середньоарифметичної фільтрації зображень при різних даних і в різних режимах. Спеціальна увага приділена перевірці працездатності методу для обробки зображень з чіткими контурами. Для таких зображень при

накладанні випадкової маски шляхом статистичної обробки виявилося можливим приблизно відновити контури зображення. Відповідно, за допомогою розробленої програми відпрацьовувалися методи створення масок, які стійкі до методів статистичної обробки зображень з етою відновлення їх контурів.

В створеній програмі передбачені різні режими фільтрації зображень, є можливість фільтрації зображень різного розміру, тобто існує можливість зміни параметрів обробки матриць. Також передбачена можливість гнучкої зміни розміру апертури зображення.

Для створення програми моделювання фільтрації зображень використано мову програмування високого рівня Java.

3.1 Організація даних програми

Для стику з діючою системою обробки аерокосмічних зображень НАНУ SPECTR 4.0, організація даних розроблених програм має відповідати організації структур даних існуючих систем обробки зображень.

Для розробки було обрано community версію середовища IntelliJ IDEA, тому що вона підтримує мову програмування Java, системи контролю версій CVS, Subversion, Mercurial і Git, засоби збирання Maven і Ant. До її складу входить модуль візуального проектування GUI-інтерфейсу Swing UI Designer, система перевірки коректності коду, система контролю за виконанням завдань.

В існуючому комплексі обробки аерокосмічних зображень SPECTR 4.0 передбачено використання сіми типів формату зображень. Найбільш часто використовується формат TIFF (Tagged Image File Format – формат файлу ознакових зображень), що забезпечує збереження та обробку чорно-білих зображень та зображень з глибиною кольору 8, 16, 24 і 32 біт високої якості та довільного розміру. При розробці програми в рамках дипломного проекту використовується модифікація формату TIFF – чорно білі

зображення з 16-ти бітовим кодом тону. В цьому форматі кожен піксель зображення представляється у вигляді 16-розрядного коду. Повне зображення форматом 1400 на 1024 пікселя представляється у вигляді матриці. Програма дозволяє згідно технічного завдання виконувати середньоарифметичну фільтрацію зображення, представленого в форматі TIFF з розмірами апертури від 9 до 17 (непарні розміри). Це робить створену програму гнучкою для різних застосувань.

Хоча прийнятий формат TIFF представлення зображень оперує з 16-розрядніми цілими кодами, для підвищення точності середньоарифметичної фільтрації виконується трансформація цих кодів в формат з плаваючою точкою. Очевидно, що при реалізації медіанної фільтрації потреби в прямому та зворотному перетвореннях типу даних немає. Це зменшує швидкодію, але при використанні арифметичного співпроцесора на кожному з процесорних блоків віддаленої комп'ютерної системи різниця в часі реалізації різних типів фільтрації не є надто помітною. Проте використання форматів з плаваючою точкою дозволяє значно зменшити помилку, яка накоплюється в процесі накладання масок на реальне зображення.

Проте всі експериментальні оцінки здійснено з урахуванням того, що найбільш часто використовуваною апертурою є квадратна апертура розміру, що обрамлює 121 елемент зображення, тобто апертура розміру 11. Такий розмір апертури дозволяє досягти якісно високого рівня усунення імпульсних завад, мінімальній втраті якості оброблюваного зображення, через втрату чіткості, шляхом розмилення контурів дрібних об'єктів.

Таким чином, в розробленій програмі дані про зображення зберігаються та оброблюються у вигляді двовимірного масиву чисел з плаваючою точкою. Числа, які представляють інтенсивність чорно-білого зображення в виконаній розробці можуть бути лише додатніми. Двовимірні масиви чисел з плаваючою точкою застосовуються в створеній програмі для

моделювання обробки зашифрованого зображення на віддаленій обчислювальній платформі.

3.2 Процедури і функції розробленої програми

Розроблені в програмі функції та процедури забезпечують виконання основних функціональних блоків шифрування, середньоарифметичної фільтрації та дешифрування зображення після повернення з віддаленої комп'ютерної системи.

Шифрування відбувається за такою схемою:

1) Кількість рядків: $(r_{\Delta 1} + r_x + 1) = 2n+1$
2) Кількість стовпчиків: $r_z = 2n-1$
3) Нумерація стовпчиків та рядків починається з нуля
4) $\forall l \in \{0,...,r_x - 1\} : M_4[l] = x \cdot 2^l$
5) $\forall k \in \{r_x,...,2n-1\} : M_4[k] = x \cdot 2^{k-r_x}$
6) $M_4[2n] = z$
7) $\forall e \in \{0,...,n-1\} : G_4[e] = d_e$
8) $\forall u \in \{n,...,2n-1\} : G_4[u] = x_{u-n}$
9) $\forall i \in \{0,...,2n\} : M_4[i][2n] = \bigoplus_{j=0}^{2n-1} M_4[i][j] \cdot G_4[i]$,

У даному випадку є відомими вектор G_4 та остання строка матриці M_4. Замість бітів x записуються їх символьні позначення.

Вважаючи на те, що додавання за модулем два двох однакових чисел дає в результаті нуль, квадрат значення біта дорівнює йому самому, вектор G_4 є відомим, то можна M_4 привести до матриці M_5, що має наступний вигляд:

1) Кількість рядків: $r_{\Delta 1} + 2 = n + 2$
2) Кількість стовпчиків: $r_z = 2n-1$

3) Нумерація стовпчиків та рядків починається з нуля

4) $\forall l \in \{0,...,r_x - 1\} : M_5[l] = x \cdot G[l] \cdot 2^l$

5) $\forall k \in \{0,...,n-1\} : M_5[n][2k] = x_k$

6) $M_5[n+1] = z$

7) $\forall i \in \{0,...,2n-2\} : M_5[i][n+1] = \bigoplus_{j=0}^{n} M_5[i][j]$

Нехай N_{M_5} - кількість стовпчиків M_5.

Алгоритм обчислення бітів x (виконання алгоритму автоматично припиняється, якщо знайдено всі біти x).

1. $\forall i \in \{0..N_{M_5} - 1\}$:

1.1. Якщо в i-тому стовпчику містяться два однакових елементи, то при додаванні за модулем два їх сума дорівнюватиме нулю. Такі елементи не впливають на результат додавання елементів стовпчику, тому повторювані елементи обнуляються.

1.2. Якщо i-тий стовпчик містить символьні позначення бітів x, що вже були обчислені, до $M_5[n+1][i]$ додаються за модулем два ті з відомих бітів x, символьні позначення яких містяться у i-тому стовпчику. Замість цих символьних позначень до стовпчика записуються нулі.

1.3. Якщо i-тий стовпчик містить єдиний ненульовий елемент (при підрахунку кількості ненульових (нульових) елементів стовпчику значення з останнього рядка не враховується) $M_5[j][i] = x_k$, де $j \in \{0..n\}$, $k \in \{0..n-1\}$, тоді:

1.3.1. зберегти значення $x_k = M_5[n+1][i]$

1.3.2. видалити i-тий стовпчик, зменшити значення N_{M_5} на одиницю

1.3.3. $i := 0$

1.4. Якщо i-тий стовпчик містить лише нульові елементи, видалити його (бо в ньому нема корисної інформації, тобто, з нього неможливо

створити лінійного рівняння для пошуку якогось біта x). Значення N_{M_5} зменшити на одиницю.

2. Обрати будь-який стовпчик (нехай його номер – i), що містить два ненульові елементи x_k та x_l, $k \in \{0..n-1\}$, $l \in \{0..n-1\}$.

2.1. зберегти значення $x_k = 1$

2.2. зберегти значення $x_l = M_5[n+1][i] \oplus 1$

2.3. видалити i-тий стовпчик, зменшити значення N_{M_5} на одиницю, повернутися до пункту 1.

3. Невідомому біту x_k, $k \in \{0..n-1\}$, присвоїти одиницю.

Для експериментальних досліджень виграшу в швидкодії також було реалізовано моделювання середньоарифметичної фільтрації зображень за допомогою функції int filtrlMeanPartial(int s, int r, int m) на обчислювальній платформі користувача. При цьому вважається, що всі обчислення виконуються на одному процесорі, тобто не виконується описаного вище розпаралелювання обчислювального процесу, як це робиться при моделюванні такої обробки на віддалених обчислювальних потужностях.

ВИСНОВКИ ДО РОЗДІЛУ 3

В результаті виконання розробок, які складають третій розділ дипломного проекту і направлені на створення програмних засобів для експериментального дослідження методу захищеної середньоарифметичної фільтрації на віддалених комп'ютерних системах доцільним є зробити такі висновки:

1. Розроблена програма моделювання методу захищеної середньоарифметичної фільтрації на віддалених комп'ютерних системах, яка дозволяє здійснювати перевірку функціональної коректності роботи методу захищеної середньоарифметичної фільтрації зображень, в тому числі і в критичних режимах його роботи, вимірювати час виконання окремих етапів обробки для експериментальної оцінки коефіцієнту прискорення середньоарифметичної фільтрації за рахунок використання віддалених високопродуктивних комп'ютерних систем.

2. З використанням розробленої програми експериментально доведено функціональну коректність методу захищеної середньоарифметичної фільтрації на віддалених комп'ютерних системах, показано, що його використання дозволяє прискорити обчислювальну реалізацію фільтрації зображень з метою видалення імпульсних завад в 10-20 раз в порівнянні з реалізацією на обчислювальній платформі користувача за рахунок того, що фільтрація на віддалених багатопроцесорних системах може виконувати паралельно над групами апертур зображення.

3. З використанням розроблених програм виконано оцінку рівня захищеності методу захищеної середньоарифметичної фільтрації на віддалених комп'ютерних системах. Показано, що метод стійкий до перебору кодів масок – їх підбір виходить за рамки технічних можливостей сучасних комп'ютерних систем.

4. З використанням розроблених програм показано, що метод захищеної середньоарифметичної фільтрації на віддалених комп’ютерних системах вразливий до спроб відновлення оригінального зображення при випадковому виборі масок. Для зображень з чіткими контурами при накладанні випадкової маски шляхом статистичної обробки виявилося можливим приблизно відновити контури зображення. Відповідно, за допомогою розробленої програми відпрацьовувалися методи створення масок, які стійкі до методів статистичної обробки зображень з метою відновлення їх контурів.

ВИСНОВКИ

В результаті виконання дипломного проекту бакалавра, направленого на розробку, теоретичну та експериментальну дослідження методу захищеної реалізації середньоарифметичної філіації на віддалених комп'ютерних системах задля прискорення цієї масової операції обробки зображень отримані наступні результати:

1. Показано, що невирішеність проблеми забезпечення інформаційної безпеки безпосередньо в процесі її обробки на віддалених і не контрольованих потужних обчислювальних систем з використанням хмарних технологій суттєвим чином стримує широке використання переваг, які надають ці технології і це педалює інтенсивні дослідження, направлені на створення ефективних засобі захисту даних користувачів при їх віддаленій обробці.

2. Для ефективного шифрування даних користувачів при їх віддаленій обробці в рамках хмарних технологій мають використовуватися гомоморфні методи: тобто процедура шифрування визначається морфологією тих перетворень над інформацією користувача, які виконуються в процесі її віддаленої обробки і вона має забезпечити можливість адекватного дешифрування результатів обробки. Гомоморфне шифрування має забезпечувати такий рівень захисту даних, щоб ресурси для подолання захисту були більшими за потенціальні вигоди від незаконного доступу до інформації користувачів. При цьому існує суттєве обмеження на об'єм ресурсів які можуть бути використані для реалізацію функцій захисту інформації: вони мають бути на порядки меншими ніж об'єм обчислювальних ресурсів, що витрачається на обробку цієї інформації.

3. До теперішнього часу створено і успішно використовується на практиці низка методів гомоморфного шифрування для задач перетворень матриць, лінійної алгебри, розпізнавання образів, модулярного експоненціювання.

4. Задачі обробки зображень являють собою одну із найбільш масових процедур сучасних інформаційних технологіях, які добре розпаралелюються і тому можуть ефективно вирішуватися на віддалених багатоядерних обчислювальних системах з залученням хмарних технологій. Більша частина зображень, які потенційно можуть оброблюватися на віддалених потужних комп'ютерних системах мають конфіденційних характер.

5. Існуючі методи захисту зображень при їх віддаленій фільтрації базуються на адитивному маскуванні і не забезпечують прийнятної для потреб сьогоднішнього дня рівня ефективності. В якості найбільш перспективних шляхів вдосконалення методів захищеної віддаленої фільтрації зображень доцільно розглядати урахування того, що на практиці частіше за все оброблюються потік зображень, які можуть перемішувати і таким чином утруднювати реконструкцію кожного конкретного зображення при їх віддаленій обробці.

6. Аналіз практики обробки аерокосмічних зображень доводить, що в більшості випадків відбувається фільтрація не одиночного зображення, а обробка потоку зображень. Це може бути використано для підвищення ефективності віддаленої захищеної фільтрації зображень в хмарах.

7. Досліджено теоретично метод групової захищеної фільтрації, в якому функції захисту реалізуються за рахунок контрольованого перемішування зображень групи. В якості секретного ключа для дешифрування виступає таблиця перемішування, яка генерується користувачем і зберігається в секреті. Показано, що підбір перестановочної таблиці на віддалених комп'ютерних системах виходить за рамки технічних можливостей сучасних комп'ютерних систем.

8. Запропонована модифікація методу захищеної віддаленої середньоарифметичної фільтрації, яка поєднує переваги групової обробки зображень з їх перемішуванням та адитивного маскування. Використання модифікованого алгоритму віддаленої середньоарифметичної фільтрації дозволяє отримати більший ефект прискорення віддаленої реалізації фільтрації.

9. Розроблена програма моделювання методу захищеної середньоарифметичної фільтрації на віддалених комп'ютерних системах, яка дозволяє здійснювати перевірку функціональної коректності роботи методу захищеної середньоарифметичної фільтрації зображень, в тому числі і в критичних режимах його роботи, вимірювати час виконання окремих етапів обробки для експериментальної оцінки коефіцієнту прискорення середньоарифметичної фільтрації за рахунок використання віддалених високопродуктивних комп'ютерних систем.

10. З використанням розробленої програми експериментально доведено функціональну коректність методу захищеної середньоарифметичної фільтрації на віддалених комп'ютерних системах, показано, що його використання дозволяє прискорити обчислювальну реалізацію фільтрації зображень з метою видалення імпульсних завад в 10-20 раз в порівнянні з реалізацією на обчислювальній платформі користувача за рахунок того, що фільтрація на віддалених багатопроцесорних системах може виконувати паралельно над групами апертур зображення.

11. З використанням розроблених програм виконано оцінку рівня захищеності методу захищеної середньоарифметичної фільтрації на

віддалених комп’ютерних системах. Показано, що метод стійкий до перебору кодів масок – їх підбір виходить за рамки технічних можливостей сучасних комп’ютерних систем.

12. З використанням розроблених програм показано, що метод захищеної середньоарифметичної фільтрації на віддалених комп’ютерних системах вразливий до спроб відновлення оригінального зображення при випадковому виборі масок. Для зображень з чіткими контурами при накладанні випадкової маски шляхом статистичної обробки виявилося можливим приблизно відновити контури зображення. Відповідно, за допомогою розробленої програми відпрацьовувалися методи створення масок, які стійкі до методів статистичної обробки зображень з метою відновлення їх контурів.

СПИСОК ЛІТЕРАТУРИ

1. Sathish V. Cloud-based Image Processing With Data Priority Distribution Mechanism./ Sathish V., Sangeetha T.A. // Journal of Computer Applications.- Vol.6, №1.- 2013.- P. 6-8.
2. Грузман И.С. Цифровая обработка изображений в информационных системах / И. С. Грузман, В. С. Киричук–Новосибирск : НГТУ, 2002.−352 с.
3. Гуменюк І.О. Метод віддаленої середньоарифметичної фільтрації зображень / І.О. Гуменюк, О.Є.Слюсаренко // Альманах науки. - 2019.- № 11 (32).- С.40-43.
4. Марковський О.П. Захищена реалізація фільтрації зображень в GRID-системах / О.П. Марковський, М.В. Невдащенко, А.М. Білашевська // Вісник Національного технічного університету України ”КПІ” Інформатика, управління та обчислювальна техніка. – Київ: ВЕК+. – 2014. – № 61. – С.105-109.
5. Форсайт Д. Компьютерное зрение. Современный подход / Д. Форсайт, Ж. Понс. – М. : Вильямс, 2004. – 928 с.
6. Шапиро Л. Компьютерное зрение / Л. Шапиро, Дж. Стокман. – М. : Бином. Лаборатория знаний, 2006. – 716 с.
7. Буйбарова М.Ф. Метод захищеної реалізації перетворень Фур’є на віддалених розподілених комп’ютерних системах / М.Ф. Буйбарова, Ю.М. Виноградов, В.Ю. Приймак // Вісник Національного технічного університету України “КПІ” Інформатика, управління та обчислювальна техніка. – К.: ТОО „ВЕК+”. – № 64. – 2016. – С. 64-71.
8. Boroujerdi N. Cloud Computing: Changing Cogitation about Computing/ N. Boroujerdi, S. Nazem // IJCSI International Journal of Computer Science Issues. – Vol. 9. – Issue 4. – 2012. – №3. – PP. 169-180.

9. Monjur Ahmed. Cloud Computing and Security Issues in the Cloud / Monjur Ahmed, Mohammad Ashraf Hossain // International Journal of Network Security PengyaoWang. Rapid processing of remote sensing images based on cloud computing /PengyaoWang,JianqinWang, YingChen, Guangyuan Ni // Future Generation Computer Systems. – Vol.29.- № 8.-2013.- pp.1963-1968.
10. Молдовян А.А. Введение в криптосистемы с открытым ключом / А.А. Молдовян, Н.А. Молдовян. – С-Пб.: БХВ-Петербург. – 2004. – 322 с.
11. Markovskyi O.P. Secure Modular Exponentiation in Cloud Systems/ O.P. Markovskyi, N. Bardis, S.J. Kirilenko // Proceeding of the Congress on Information Technology. Computationnal and Experimental Physics (CITCEP 2015), 18-20 December 2015, Krakow. Poland. – PP.266-269.
12. Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу. НД ТЗІ 2.5-004-99.
13. Вельшенбах М. Криптография на С и С++ в действии / М. Вельшенбах. – М.: Триумф. – 2004. – 460 с.
14. Sutton M. A. Image Correlation for Shape, Motion and Deformation Measurements (Basic Concepts, Theory and Applications)./ Sutton M. A., Orteu J. J., Schreier H. // – New York: Springer, 2009. – 364 p.
15. Malgouyres F. A noise selection approach of image restoration, Applications in signal and image processing IX / F. Malgouyres // Vol 4478. – 2001. – P. 34-41.
16. Таненбаум Э. Компьютерные сети. 6-е изд / Э. Таненбаум. – М.:Питер. – 2013. – 991 с.
17. Макконелл Д.Х. Основы современных алгоритмов / Д.Х. Макконелл. – М.: Вильямс. – 2004. – 512 с.

18. Харин Ю.С. Математические и компьютерные основы криптологии / Ю.С. Харин, В.И. Берник, Г.В. Матвеев, С.В. Агиевич. – Мн.: Новое знание. – 2003. – 382 с.

19. Марковський О.П. Метод прискореної захищеної фільтрації зображень на віддалених комп'ютерних системах / О.П. Марковський, І.О. Гуменюк., Міратаї Аліреза, Я.І. Торошанко, М.О.Волощук // Телекомунікаційні та інформаційні технології**.** - № 4 (65).- 2019.- С.99-110.

20. Янтуш Д. А. Дешифрирование азрокосмических снимков / Д. А. Янтуш. – М. : Недра, 1991. – 240 с.

21. Трифонов Т. А. Геоинформационные системы и дистанционное зондирование в экологических исследованиях / Т. А. Трифонов. − М. : Академический проект, 2005. − 252 с.

22. Ростовцев А.Г. Алгебраические основы криптографии / А.Г. Ростовцев. – СПб.: Мир и семья. – 2000. – 353 с.

23. Сэвидж Д.Э. Сложность вычислений / Д.Э. Сэвидж. – М.: Факториал. – 1998. – 368 с.

24. Брэй Б. Микропроцессоры Intel. Архитектура, программирование и интерфейсы. Восьмое издание / Б. Брэй; пер. с анг. А.В.Жукова.- Санк-Петербург: БХВ-Петербург, 2015.-1328 с.

25. Лисицин В. З. Практикум по фотограмметрии и дистанционному зондированию / В. З. Лисицин. – Харьков : ХНАГХ, 2006. – 200 с.

26. Савиных В. П. Аэрокосмическая фотосьемка / В. П. Савиных, А. С. Кучко, А. Ф. Стеценко. – М. : КартоГеоЦентр Геоиздат, 1997. – 378 с.

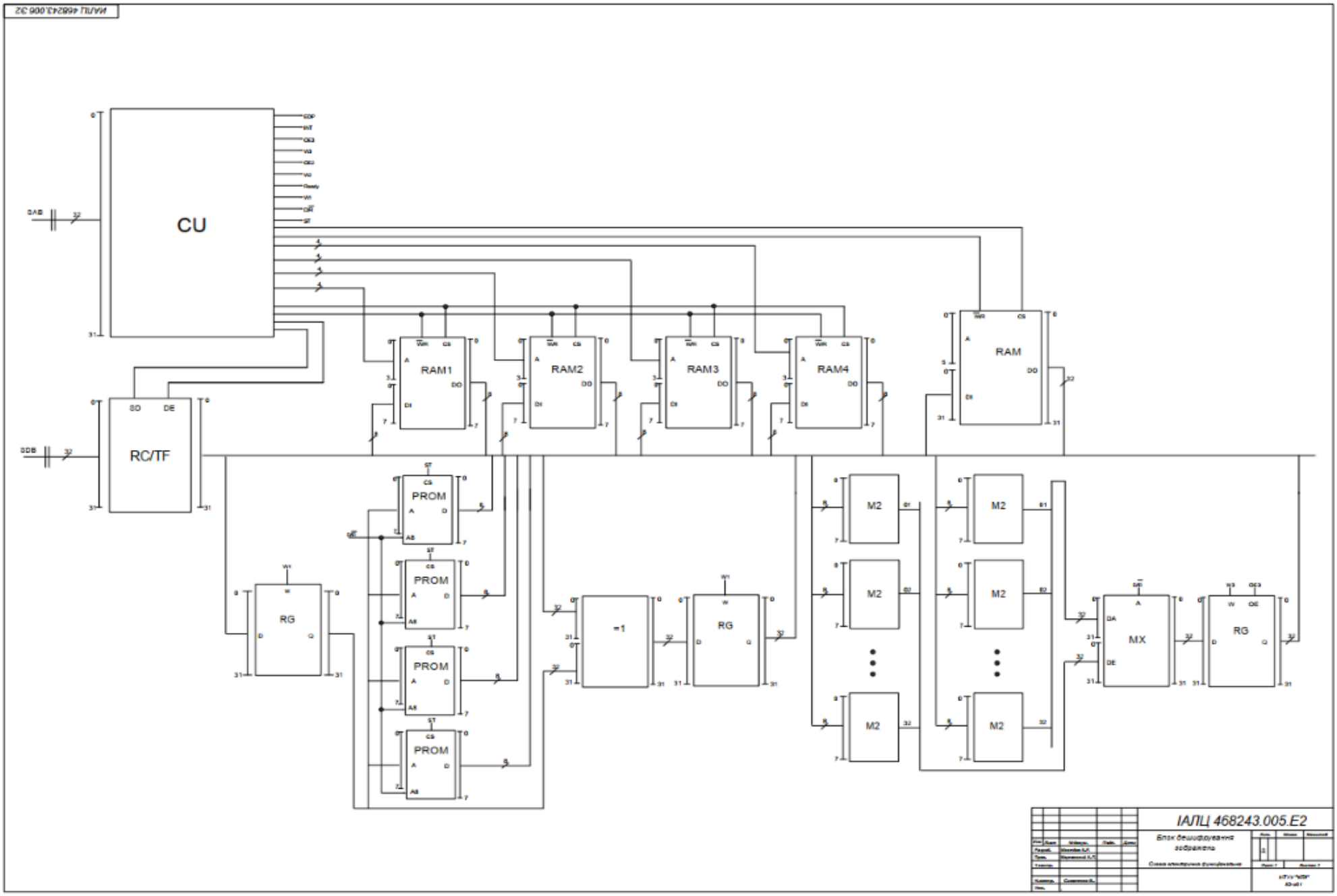
27. Березин А.С. Защита информации в открытых сетях / А.С. Березин, С.А. Петренко // Корпоративные системы. – 2001. – № 2. – С.65-69.

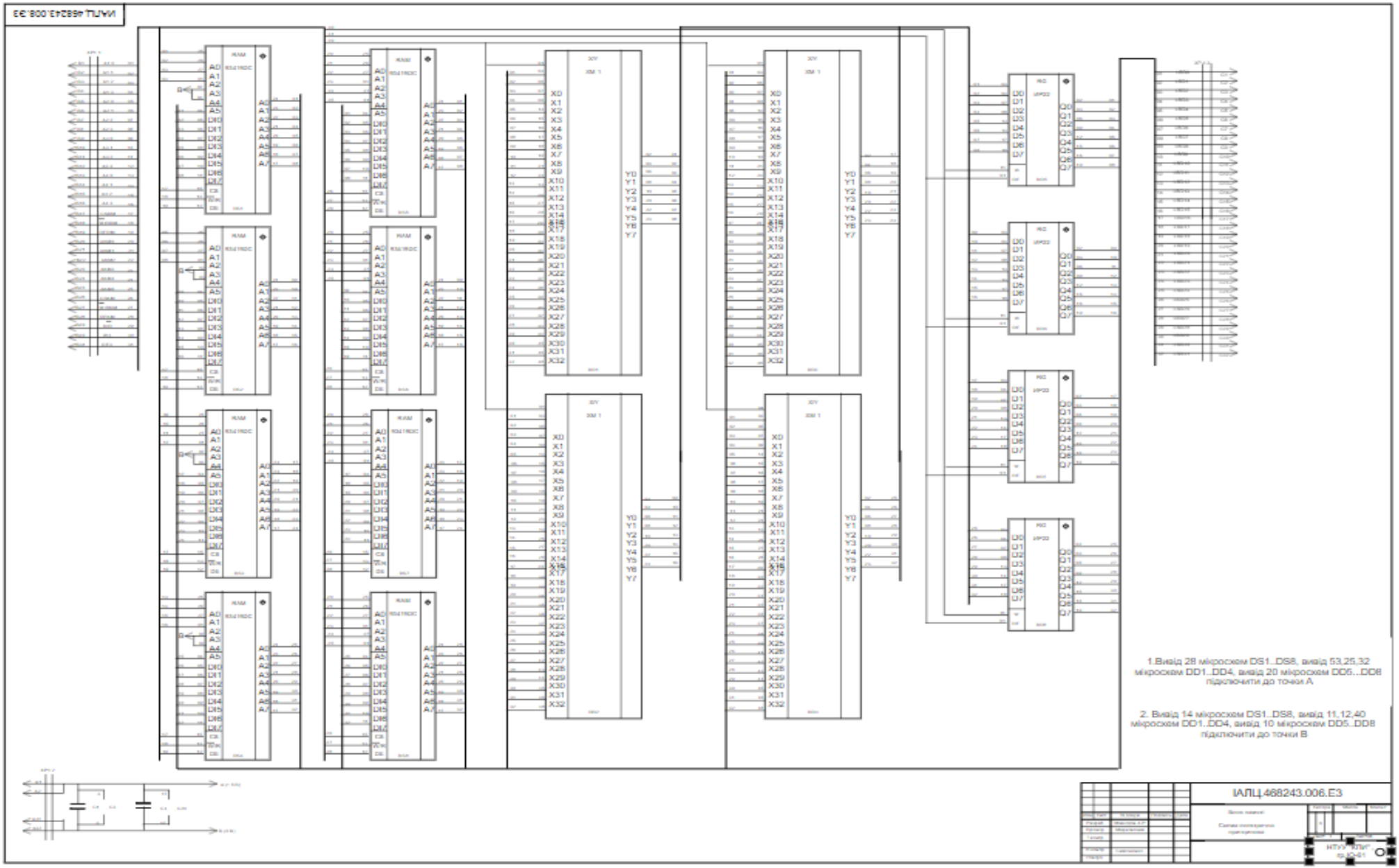
28. Гамаюн В.П. Квазиграфический метод преобразования многоразрядного кода / В.П. Гамаюн // Комп'ютерні засоби, мережі та системи. Зб.наукових праць. – К.: Ін-т кібернетики ім.В.М.Глушкова НАНУ. – 2002. – №1. – С.53-57.

29. Зима В.М. Безопастность глобальных сетевых технологий / В.М. Зима, А.А. Молдовян, Н.А. Молдовян. – СПб.: БХВ-Петербург. – 2002. – 320 с.

30. Иванов М.А. Криптографические методы защиты информации в компьютерных системах и сетях / М.А. Иванов. – М.:Кудиц-Образ. – 2001. – 368 с.

Додатки

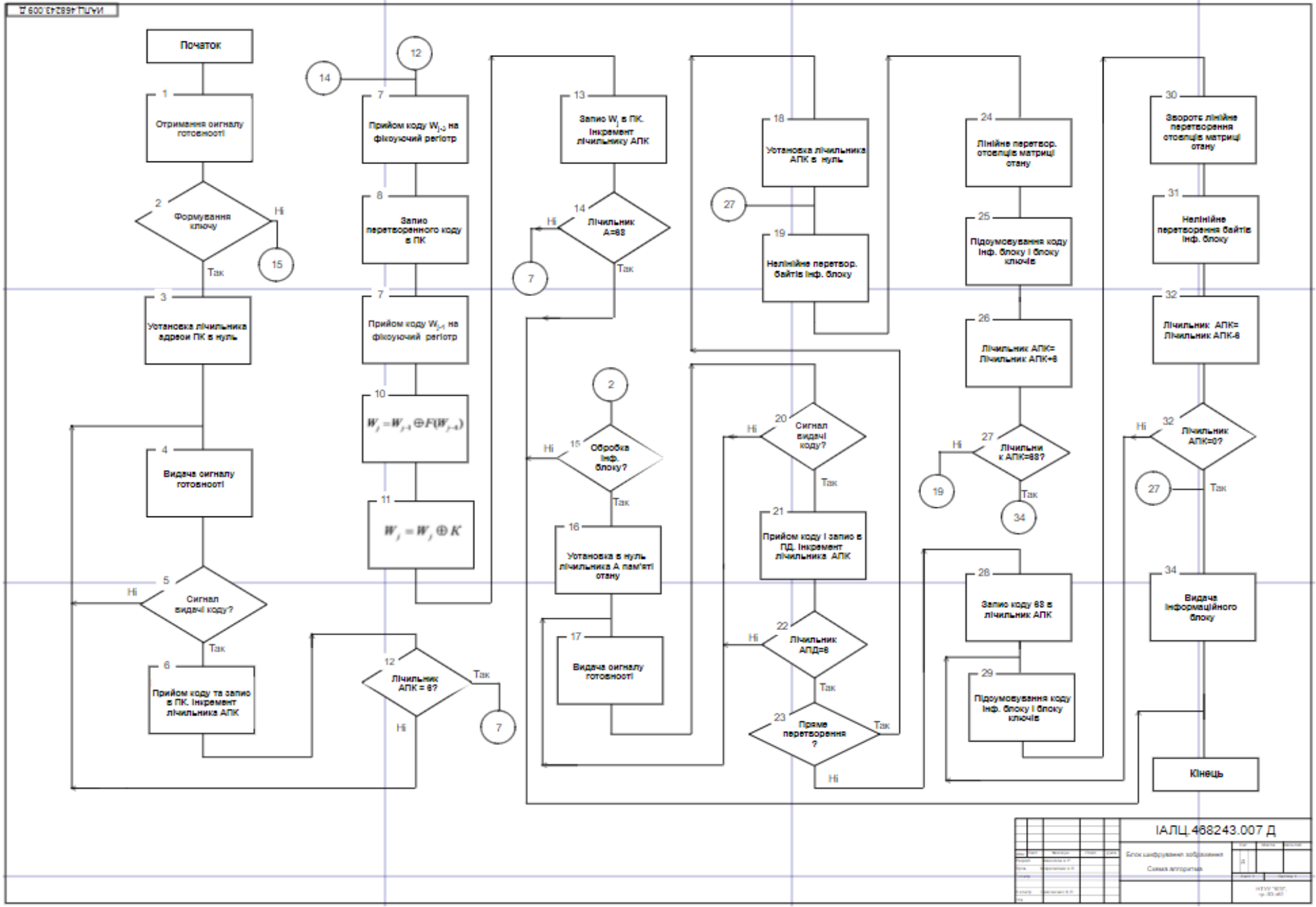

ІАЛЦ 468243.005 Е2
CU
RC/TF
RAM1
RAM2
RAM3
RAM4
RAM
PROM
PROM
PROM
PROM
RG
RG
RG
M2
M2
M2
M2
M2
M2
MX
ІАЛЦ 468243.005 Е2
Блок дешифрування зображення



1.Вивід 28 мікросхем DS1...DS8, вивід 53,25,32 мікросхем DD1...DD4, вивід 20 мікросхем DD5...DD8 підключити до точки А

2. Вивід 14 мікросхем DS1...DS8, вивід 11,12,40 мікросхем DD1...DD4, вивід 10 мікросхем DD5...DD8 підключити до точки В

ІАЛЦ.468243.006.Е3


ІАЛЦ.468243.009 Д
Початок
1
Отримання сигналу готовності
2
Формування ключу
Ні
Так
15
3
Установка лічильника адреси ПК в нуль
4
Видача сигналу готовності
5
Сигнал видачі коду?
Ні
Так
6
Прийом коду та запис в ПК. Інкремент лічильника АПК
12
14
7
Прийом коду W_{j-2} на фіксуючий регістр
8
Запис перетвореного коду в ПК
7
Прийом коду W_{j-1} на фіксуючий регістр
10
$W_j = W_{j-1} \oplus F(W_{j-4})$
11
$W_j = W_j \oplus K$
12
Лічильник АПК = 8?
Ні
Так
7
13
Запис W_j в ПК. Інкремент лічильнику АПК
14
Лічильник А=88
Ні
Так
7
2
15
Обробка інф. блоку?
Ні
Так
16
Установка в нуль лічильника А пам'яті отану
17
Видача сигналу готовності
18
Установка лічильника АПК в нуль
27
19
Нелінійне перетвор. байтів інф. блоку
20
Сигнал видачі коду?
Ні
Так
21
Прийом коду і запис в ПД. Інкремент лічильника АПК
22
Лічильник АПД=8
Ні
Так
23
Пряме перетворення ?
Так
Ні
24
Лінійне перетвор. отовпців матриці отану
25
Підсумовування коду Інф. блоку і блоку ключів
26
Лічильник АПК= Лічильник АПК+8
27
Лічильни к АПК=88?
Ні
Так
19
34
28
Запис коду 88 в лічильник АПК
29
Підсумовування коду Інф. блоку і блоку ключів
30
Зворотє лінійне перетворення отовпців матриці отану
31
Нелінійне перетворення байтів Інф. блоку
32
Лічильник АПК= Лічильник АПК-8
32
Лічильник АПК=0?
Ні
Так
27
34
Видача Інформаційного блоку
Кінець
ІАЛЦ.468243.007 Д
Блок шифрування зображення
Схема алгоритма

Додатки. Лістинги програми моделювання захищеної реалізації середньоарифметичної фільтрації на віддалених комп'ютерних системах

```
#include<stdio.h>
#include<fstream.h>
#include<conio.h>
#include<iostream.h>
#include<dos.h>
#include<snap.h>
#include <math.h>
#include <stdlib.h>
#include <time.h>

unsigned long UG(unsigned long,unsigned long, unsigned long);
unsigned long EG(unsigned long,unsigned long, unsigned long);
unsigned long UMO(unsigned long,unsigned long, unsigned long, int n);
unsigned long UMG(unsigned long,unsigned long, unsigned long, int n);
unsigned long CC(int,int);
int HE(int,int);

int main()
{
  long n,e,m,h,k,j,i,q,u,r,t,mod,ste,v,v_a,v_b;
  double a,M,p,b,d,c,s,g,z,l_av,sigma,gamma,pp,dt;
  cout<<"\n";
  long A[1000];
  long L[1000];
  long B[1000];
  long F[1000];
  long B_D[1000];
  long F_D[1000];

  ofstream FA("INNA_51.TXT",ios::out);

  m=1000;  a=0.02; b=1.0;
  M=m/a; d= 1.0/a;
  v=1.2*l_av; v=100;
  l_av=100; sigma = l_av/6;
  a=0.001;  gamma = 0.5;
  v_a= gamma*v;   v_b=v - v_a;

  FA<<"\n Average lenght   450"<<"  Alfa   0.001  Number of key   10000";
  do
  {
    M=m/a;
    pp=0.0;
    for(j=0;j<m;j++) A[j]=rand()*M/RAND_MAX;
    do
    {
```

```
  e=0;
  for(j=0;j<m-1;j++)
  if(A[j]>A[j+1]) { t=A[j]; A[j]=A[j+1]; A[j+1]=t; e=1; }
}
while(e);
A[m]=M;
for(j=0;j<m;j++)
{
  c=0; for(i=0;i<12;i++) c+=(0.0+rand())/RAND_MAX;
  c=(c-6)*sigma+l_av; if(c<=1) c=1;
  L[j]=c;
}
 // RECORDING
r=m;
if( (v_a-A[0]) > 0 ) B[0]=0; else B[0]=A[0]-v_a;
F[0]=B[0]+L[0];
B_D[0]=0; F_D[0]=0;
if(F[0]>=A[1])
{
  B_D[0]=0;
  F_D[0]=F[0]-A[1];
  F[0]=A[1]-1;
  r--;
}
else
if(F[0]>A[0]+v_b)
{
  B_D[0]=0;
  F_D[0]=F[0]-(A[0]+v_b);
  F[0] = A[0]-v_b;
  r--;
}
dt=F_D[0];
pp+=(L[0]-dt)/L[0] +2*dt/L[0];
for(j=1;j<m;j++)
{
  h=A[j]-v_a;
  if(h>F[j-1]) B[j]=h; else  B[j]=F[j-1]+1;
  F[j]=B[j]+L[j];
  B_D[j]=F_D[j-1]; F_D[j]=F_D[j-1];
  if(F[j]>=A[j+1])
  {
  B_D[j]++;
  F_D[j]=B_D[j]+F[j]-A[j+1]+1;
  F[j]=A[j+1]-1;
  r--;
  }
  else
  if(F[j]>A[j]+v_b)
  {
  B_D[j]=F_D[j-1]+1;
  F_D[j]=B_D[j]+ F[j]-(A[j]+v_b);
```

```
    F[j] = A[j]+v_b;
    r--;
    }
    if(j<m-1)
    {
    dt=F_D[j]-B_D[j];
    dt=(L[j]-dt)/L[j] +2*dt/L[j];
    pp+=dt;
    }
 }
for(j=0;j<6;j++) D4[j]=D5[j];
i=6;
for(j=1;j<64; j++)
if(NUE(j)==4) D4[i++]=j;
ofstream FA("MARY_10.TXT",ios::out);
cout<<"\n i="<<i;
f=0;  k=0;
for(i=0;i<19;i++)
for(l=i+1;l<20; l++)
for(h=l+1;h<21;h++)
{

   B[0]=D4[i]; B[1]=D4[l]; B[2]=D4[h];
   q=2;
   S[0]=NUE(B[0]^B[1]);       if(S[0]==2) q--; else if(S[0]<2) q=0;
   S[1]=NUE(B[0]^B[2]);       if(S[1]==2) q--; else if(S[1]<2) q=0;
   S[2]=NUE(B[1]^B[2]);       if(S[2]==2) q--; else if(S[2]<2) q=0;
   S[3]=NUE(B[0]^B[1]^B[2]);  if(S[3]==2) q--; else if(S[3]<2) q=0;

   if(q>0)
   {
    f=1; k++;
    FA<<"\n "<<"  "<<hex<<B[0]<<"  "<<B[1]<<"   "<<B[2];
    cout<<"\n "<<hex<<B[0]<<"  "<<B[1]<<"  "<<B[2]<<"   "<<l;
    cout<<"    s="<<S[0]<<"  "<<S[1]<<"  "<<S[2]<<"   "<<S[3];
     //   FA<<" d="<<S[0]<<"  "<<S[1]<<"  "<<S[2]<<"  "<<S[3];
     //   FA<<" =  "<<C[0]<<"  "<<C[1]<<"   "<<C[2];
     //   if(u2>=0) FA<<"  -> "<<dec<<u2;
   }
}
cout<<"\n f="<<f<<"  k= "<<dec<<k;
/*
ofstream FA("MARY_7_B.TXT",ios::out);
cout<<"\n Alila\n";
unsigned long n,m,f,h,k,j,q,d,u,r,t,mod,ste,l,v,w,y;
int D[26]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,20,21,22,24,26,28,30};
int A[40]={31,15,23,27,29,0x3,0x16,0x11};
int B[40];
int C[20];
int i2,i3;
float a,ama,ami;
for(i2=0;i2<85; i2++)
```

```
{
  A[5]=T[i2][0]; A[6]=T[i2][1]; A[7]=T[i2][2];
  w=0; h=0; v=0;
  h=0; v=0;
  for(n=0;n<7;n++)
  for(m=n+1;m<8; m++)
  {
    i=0;
    for(j=0;j<8; j++) if( (j!=n)&&(j!=m)) B[i++]=A[j];
    h++; f=0;
    for(q=0;q<6;q++)
    {
    i=0;
    for(j=0;j<6;j++) if( j!=q ) C[i++]=B[j];
    if(ORT(&C[0],5)) f=1;
    }
    if(f) v++;
  }
  a=h;
  a= v/a;
 b=(r+0.0)/m;
for(i=0;i<8;i++)
for(j=0;j<8;j++)
A[i][j]=cos(3.14*i*j*0.25)/8;
for(i=0;i<8;i++)
{
  FA<<"\n";
  for(j=0;j<8;j++)
  FA<<A[i][j]<<"  ";
}
FA<<"\n";
for(i=0;i<8;i++)
for(j=0;j<8;j++)
C[i][j]=sin(3.14*i*j*0.25)/8;
for(i=0;i<8;i++)
{
  FA<<"\n";
  for(j=0;j<8;j++)
  FA<<C[i][j]<<"  ";
}
for(i=0;i<8;i++)
{
   R[i]=0; M[i]=0;
   for(j=0;j<8;j++)
   {
    R[i]+=X[i]*A[i][j];
    M[i]+=X[i]*C[i][j];
   }
}
FA<<endl;
for (j=0;j<8;j++) FA<<X[j]<<"  ";
FA<<endl;
```

```
  for (j=0;j<8;j++) FA<<R[j]<<"  ";
  FA<<endl;
  for (j=0;j<8;j++) FA<<M[j]<<"  ";
  for(i=0;i<8;i++) D[i]=X[i]-B[i];
  for(i=0;i<8;i++)
  {
     U[i]=0; W[i]=0;
     Z[i]=0; Y[i]=0;
     for(j=0;j<8;j++)
     {
      U[i]+=B[i]*A[i][j];
      W[i]+=B[i]*C[i][j];
      Z[i]+=D[i]*A[i][j];
      Y[i]+=D[i]*C[i][j];
     }
  }
  FA<<endl;
  for (j=0;j<8;j++) FA<<B[j]<<"  ";
  FA<<endl;
  for (j=0;j<8;j++) FA<<U[j]<<"  ";
  FA<<endl;
  for(j=0;j<8;j++) FA<<W[j]<<"  ";
  FA<<"\n  D, Z, Y ";
  FA<<endl;
  for (j=0;j<8;j++) FA<<D[j]<<"  ";
  FA<<endl;
  for (j=0;j<8;j++) FA<<Z[j]<<"  ";
  FA<<endl;
  for(j=0;j<8;j++) FA<<Y[j]<<"  ";

  for(i=0;i<8;i++)
  {
     R[i]=U[i]+Z[i];
     M[i]=W[i]+Y[i];
  }
  FA<<"\n R and M";
  FA<<endl;
  for (j=0;j<8;j++) FA<<R[j]<<"  ";
  FA<<endl;
  for(j=0;j<8;j++) FA<<M[j]<<"  ";
   FA<<"\n "<<a<<"          "<<b<<"         "<<pp/m<<"
"<<(F_D[m-1]-1.0)/M;
   //FA<<"
"<<(m*(l_av+1))/(M+F_D[m-1]);
   //FA<<"\n a "<<a<<" % in hash-memory only "<<(r+0.0)/m;
   //FA<<" V_DOP = "<<(F_D[m-1]+0.0)/M;
   //for(j=0; j<m; j+=1) FA<<"\n "<<j<<"     "<<A[j]<<"   "<<L[j]<<"
beg="<<B[j]<<"  fin="<<F[j]<<"  db="<<B_D[j]<<"  "<<F_D[j];
   a+=0.001;
  }
  while(a<0.016);
```

```
n=0;
  for(i1=1;i1< 59; i1++)
  if (HE(i1,0)>=d)
  for(i2=i1+1;i2<60; i2++)
  if(HE(i2,0)>=d)
  for(i3=i2+1;i3<=61;i3++)
  if(HE(i3,0)>=d)
  for(i4=i3+1;i4<=62;i4++)
  if(HE(i4,0)>=d)
  for(i5=i4+1;i5<=63;i5++)
  if(HE(i5,0)>=d)
 // for(i6=i5+1;i6<=62;i6++)
 // for(i7=i6+1;i7<=63;i7++)
  {
    s=1;
    if(HE(i1,i2)<d) s=0;
    if(HE(i1,i3)<d) s=0;
    if(HE(i1,i4)<d) s=0;
    if(HE(i1,i5)<d) s=0;

    if(HE(i2,i3)<d) s=0;
    if(HE(i2,i4)<d) s=0;
    if(HE(i2,i5)<d) s=0;

    if(HE(i3,i4)<d) s=0;
    if(HE(i3,i5)<d) s=0;

    if(HE(i4,i5)<d) s=0;

  FA.close();
  return 0;
}
unsigned long UMG(unsigned long a,unsigned long b, unsigned long m, int n)
{
  int j;
  unsigned long y=0;
  for(j=0;j<n;j++)
  {
    if(b&1) y^=a;
    if(y&1) y^=m;
    y>>=1;
    b>>=1;
  }
  return y;
}

unsigned long UMO(unsigned long a, unsigned long b, unsigned long m, int n)
{
  unsigned long r=0; int k=n;
  while(n>0)
  {
    if(a&1) r+=b;
```

```
    if(r&1) r+=m;
    r>>=1;
    a>>=1;
    n--;
  }
  a=1;
  for(b=1;b<=k;b++) a<<=1;
 //  a=a%m;
 // r=(a*r)%m;
  return r;
}
unsigned long UG(unsigned long a,unsigned long b, unsigned long m)
{
  unsigned long i,j,r,k;
  r=0;
  while(b!=0)
  {
    if(b&1) r^=a;
    a<<=1;
    b>>=1;
  }
  if(m==0) return r;
  i=0x80000000;
  do
  {
    while((r&i)==0) i=(i>>1)&0x7FFFFFFF;
    j=m; k=0;
    if(j<i)
    while((j&i)==0) { j<<=1; k++; }
    if(k>0) r^=j;
  }
  while(k>0);
  j=0x40000000;
  while((m&j)==0) j>>=1;
  if(r>=j) r^=m;
  return r;
}

unsigned long EG(unsigned long a, unsigned long e, unsigned long m)
{
   unsigned long i,j,k,r;
   r=1;
   for(j=1;j<=e;j++)
   r=UG(a,r,m);
   return r;
}
int HE(int k,int m)
{
  int j=0;
  k^=m;
  while(k>0)
  {
```

```
    j+= k&1;
    k>>=1;
  }
  return j;
}

unsigned long CC(int n, int k)
{
  unsigned long i,j;
  j=1;
  for(i=1;i<=k;i++)
  j=(n-i+1)*j/i;
  return j;
}
```